BAYESIAN OPTIMIZATION OF VISUAL COMFORT

THÈSE N^{O} 3918 (2007)

PRÉSENTÉE LE 15 NOVEMBRE 2007

À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE D'ÉNERGIE SOLAIRE ET PHYSIQUE DU BÂTIMENT
PROGRAMME DOCTORAL EN ENVIRONNEMENT

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

David LINDELÖF

ingénieur physicien diplômé EPF
et de nationalité suédoise

acceptée sur proposition du jury:

Prof. A. Mermoud, président du jury
Prof. J.-L. Scartezzini, Dr N. Morel, directeurs de thèse
Prof. L. Halonen, rapporteur
Prof. S. Morgenthaler, rapporteur
Dr T. Schumann, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2007

Now *Faithful* play the Man, speak for thy God:
Fear not the wicked's malice, nor their rod:
Speak boldly man, the Truth is on thy side;
Die for it, and to Life in triumph ride.

John Bunyan, *The Pilgrim's Progress*

Abstract

We propose a self-commissioning, user-adaptive blinds and electric lighting controller for small office rooms. Self-commissioning, in this context, means that the controller builds an internal representation of the room, in particular of the room's daylighting characteristics, automatically and without user input. By user-adaptive, we mean that the illuminances the controller will seek to maintain are derived from a statistical analysis of the user's behaviour on the manually overridable blinds and electric lighting.

Self-commission and user-adaptation are implemented by two decoupled software elements.

The first element is a method for modeling the daylighting illuminance on arbitrary locations in the office room, when the windows are shaded by one or two venetian blinds (though the method can be generalized to an arbitrary number and kinds of window shadings). It uses the past history of illuminance distributions in the office room for a similar scene configuration, and models the current illuminance on a given point as a linear combination of outdoor global and diffuse irradiance.

The second element is an algorithm for the estimation of the user's visual discomfort probability. It is a function of the current illuminance distribution in that office room, and of the past history of the user's interactions with the blinds' and lighting controls. A bayesian formalism is applied to infer the probability that any illuminance distribution should be considered by the user as visually uncomfortable.

We describe how these elements have been integrated in a blinds and electric lighting controller. That controller runs today on an office room of the experimental LESO building and we present the results of the algorithm's adaptation to the preferences of that room's user.

We have also assessed that controller's performance on computer-simulated virtual office rooms. We have let the controller run for one year simulated time on six different combinations of office room location (Rome and Brussels) and orientation (north, west and south). These simulations have let us evaluate the energy savings made possible with such a controller, and the improvement of the user's visual comfort.

Keywords: Bayes's theorem, daylighting controller, user adaptation, self-commissioning, smart buildings, embedded controller, non-parametric density estimation, linear daylighting model.

Résumé

Nous proposons un système de commande automatique de stores vénitiens et de l'éclairage électrique pour des bureaux individuels. Ce système s'adapte, d'une part, aux caractéristiques lumineuses du bureau concerné; et d'autre part, aux préférences de l'utilisateur, en choisissant des niveaux d'éclairement en fonction d'une analyse statistique du comportement de l'utilisateur sur les commandes mises à sa disposition.

Deux modules logiciels découplés permettent la double adaptation aux caractéristiques du bureau et aux préférences de l'utilisateur.

Nous décrivons ces deux modules qui constituent ce système de commande. Dans un premier temps nous décrivons une méthode de prédiction de l'éclairement naturel sur des points arbitraires dans un bureau, pour des fenêtres munies d'un ou deux stores vénitiens (la méthode se généralise facilement à un nombre arbitraire de protections solaires). La méthode consiste à considérer l'éclairement en un point donné, et pour une configuration de scène donnée, comme une combinaison linéaire de l'irradiance globale et diffuse extérieure.

Dans un deuxième temps, nous décrivons un algorithme pour l'estimation de la probabilité d'inconfort visuel de l'utilisateur. Fonction de la distribution de l'éclairement dans la pièce, la méthode utilise le formalisme bayesien pour analyser les situations ayant par le passé provoqué une intervention manuelle de l'utilisateur sur ses stores ou l'éclairage électrique. De cette analyse, une probabilité d'inconfort visuel peut être dérivée pour toute distribution d'éclairement.

Ces deux éléments ont été intégrés dans un régulateur de stores vénitiens et d'éclairage artificiel. Ce régulateur commande aujourd'hui les stores et l'éclairage dans un bureau du bâtiment experimental LESO. Nous présentons le résultat de l'adaptation de ce régulateur aux préférences de l'occupant de ce bureau.

Nous avons évalué les performances énergétiques de ce système par simulations informatiques. Six combinaisons de bureaux, à savoir deux emplacements (Rome et Bruxelles) et trois orientations (nord, ouest et sud), ont été simulées en présence du régulateur. Ces simulations ont permis d'évaluer les économies d'énergie possibles avec notre système, ainsi que les améliorations du confort visuel.

Mots-clés: Théorème de Bayes, commande de l'éclairage naturel, adaptation à l'utilisateur, mise en service automatique, commande embarquée, estimation non-paramétrique de densité, modèle linéaire d'éclairage naturel.

Foreword

> I've been told that when people finish their doctoral studies, they rarely are neutral about their topic. Some are sick of the topic and quickly move on to something else. Others remain enthusiastic about the topic. I was in the latter camp.
>
> *(Martin Fowler)*

To paraphrase Tolkien, this work grew in the writing.

What we initially thought would be a trivial modification of existing blinds control algorithms to handle an arbitrary number of venetian blinds ended up as a very different beast. A log entry on 17 September 2003, in the first of four logbooks filled during this work, records the first tentative explorations of a bayesian analysis of user behaviour—what ended up as the core of this thesis and a published paper.

By the time I wrote my thesis research plan in May 2003, it was clear our bayesian controller would need a companion software module for modeling daylighting illuminance from an arbitrary number of venetian blinds. At the time we still thought it would be possible to measure in-situ the daylight coefficients for an arbitrary office room. The simplicity and robustness of the daylighting model that was eventually developed—at that time, a polynomial model in the outdoor global irradiance—took us by surprise.

A further twist came, according to my logbooks, in February 2006 when our partners insisted that the outdoor diffuse irradiance be "somehow" taken into account in the daylighting model. I initially resisted this idea, until it became clear that it would lead to a linear daylighting model instead of a polynomial one—something nobody had anticipated.

Serendipity is defined by the American Heritage Dictionary as the faculty of making fortunate discoveries by accident. Serendipity has characterized much of this work—the previous existence of vast data archives on which to test the bayesian model, the unforeseen benefits of including the diffuse outdoor irradiance in the daylighting model—but never as much as when we decided to build a simulation framework for testing the controller. Accidental design decisions made years before I joined LESO-PB made it possible to develop a simulation framework in mere weeks—the first tests started in July 2006. It is unclear whether this thesis would have been possible without this tool.

What you read today is far removed from what I wrote three years ago in my thesis research plan, a mandatory three-year plan that EPFL requires of all doctoral students after one year of enrollment. With the benefit of hindsight, any research plan that stretches out more than one year—or even one month—into the future should be eyed with skepticism. Research that proceeds according to plan for three years, without surprises or accidental discoveries, was probably not worth doing in the first place. A mandatory research plan was introduced by EPFL in 2003, and I hope this first batch of doctoral graduates (of which I am part) will provide enough constructive feedback to amend this regulation. But I digress.

The following people have each contributed in their way to this work and I would like to thank them:

Prof Jean-Louis Scartezzini and Dr Nicolas Morel, my thesis advisors, for granting me enough freedom and independence to pursue this research,

Dr Antoine Guillemin, my predecessor at LESO-PB, for his patience and efforts to ensure I could take over his work seamlessly,

Jessen Page, with whom I had the pleasure to share my office room for the last two years, and with whom I have had many an interesting and enlightening discussion (not to mention fine chess games),

The collaborators on the Ecco-build project, during which most of the ideas presented in this work germinated, in particular Christophe Marty, Dr Sif Khénioui, Dr Marc Fontoynont, Jan Wienold, Tilmann Kuhn and Dr Jens Christoffersen,

Dr Darren Robinson and Lee Ann Nicol, for patiently proof-reading two of my early papers. It was Lee who first brought to my attention the virtues of non-sexist writing,

Irmeli Svendsen and my beloved Christine, who have patiently proof-read the "un-technical" parts of this manuscript and given me the precious opinion of laypersons,

The support staff of the LESO-PB: Laurent Deschamps, Pierre Loesch, Suzanne Leplattenier, Sylvette Renfer and Barbara Smith, each of whom I have at some point or another pestered with completely unreasonable requests,

Dr Arne Kovac, who helped me solve problems in his implementation of the taut-string algorithm,

Dr Sylvain Sardy, who provided me with numerous advice with some of the most mathematically tricky questions,

Dr Paul Murrel, who showed me how to embed Computer Modern fonts in plots produced by R,

Yannick Wurm, for his clarifications on some biological aspects of visual comfort,

Malcolm Reynolds, whose witticisms made the last months spent writing this manuscript lighter to bear,

The Swiss Federal Office of Education and Research for funding this work,

All the programmers and engineers who have contributed the open source software without which this work would have been impossible, in particular the Free Software Foundation, the R Foundation for Statistical Computing, the Linux Project, Richard Stallmann, Prof Donald E. Knuth and Dr Leslie Lamport,

All the good people at LESO-PB, too numerous to list here in full, who all contributed to a warm and friendly atmosphere,

and last, but far from least, all my gratitude to Christine, who encouraged me to embark on this journey. Thank you for your optimism and your patience. I love you, Christine, and apologize for all the evenings you had to spend alone.

Contents

1 Results summary

> Where there are so many, all speech becomes a debate without end. But two together may perhaps find wisdom.
>
> *(J. R. R. Tolkien, The Fellowship of the Ring)*

Bayesian statistics—the same formalism driving most modern spam filters—can be used to build better automatic building management systems.

When you receive unwanted spam in your inbox, chances are your email program provides a button that lets you send that spam to a junk folder. Your other messages are assumed by the program to be normal email. A module in your email program then silently runs a statistical analysis, both on your junk email and on your normal email, in order to improve its capacity to classify directly incoming email as spam or ham.

We have built an integrated venetian blinds and electric lighting control system that runs according to the similar principles. When a office room user is sufficiently disturbed by insufficient lighting or glare to act on the blinds or electric lighting controls at their disposal, our system learns that the visual environment before that reaction was visually uncomfortable. Similarly, the environment after adjustment by the user is learned by the system as being supposedly comfortable.

Bayesian statistics are used by our controller to analyze this data, and to estimate in advance the probability that any scene configuration will be visually uncomfortable. At regular intervals, our controller uses its internal daylighting model of the controlled office room to calculate whether the visual discomfort probability could be reduced by adjusting the venetian blinds or the electric lighting. If that probability can be significantly reduced by adjusting the blinds or the electric lighting, then the appropriate commands are sent to the blinds and lighting actuators.

We have also installed our controller on an occupied office room of the LESO experimental building. Our controller has learned from the behaviour of that room's occupant his visual preferences, and uses that data to control the venetian blinds and the electric lighting in an optimal way—both from the user's point of view and from an energetic point of view.

Provided the user's visual discomfort probability is kept reasonably small, the controller also optimizes the use of free solar gains in order to reduce the office room's heating or cooling loads.

Computer simulations have shown that compared with a manual operation, our controller achieves, on average:

- 60% energy savings on electric lighting,
- up to 35% energy savings on heating/cooling, but with a strong dependency on office location and orientation,
- between 11% and 40% energy savings on the total energy demand,

- a drop on the yearly average visual discomfort probability from 0.44 to 0.33.

The controller running on the real, occupied LESO office room has resulted in a reduction by half of the rate of user interactions, suggesting the system was accepted by that user.

Throughout this work we have tried to keep in mind a possible industrial implementation of the ideas underlying our controller. The latter was implemented in a popular programming language, available on many embedded platforms. And the building automation system of the LESO building is not custom-built, but commercially available. Our controller software is ready to be deployed on any platform. We have tried to make the technology developed in this work readily transferable to the industry, and believe it is.

2 The need for integrated daylighting controllers in modern buildings

> The traditional way to begin talking about something is to outline the history, broad principles and the like. When someone does that at a conference, I get slightly sleepy. My mind starts wandering with a low-priority background process that polls the speaker until he or she gives an example. The examples wake me up because it is with examples that I can see what is going on. With principles it is too easy to make generalizations, too hard to figure out how to apply things. An example helps make things clear.
>
> *(Martin Fowler)*

In this chapter we will ask ourselves why an integrated daylighting and electric lighting controller (herafter "daylighting controller") is needed in today's buildings, and what requirements it must fulfill. In section 2.1 we begin by reviewing the important contribution to carbon emissions from the buildings sector. Section 2.2 will review why manual control is not sufficient to achieve a correct management of daylighting. Section 2.3 is a brief review of the recently acknowledged health benefits of daylight. In section 2.4 we will explore the current state-of-the-art in estimating or evaluating visual discomfort. Section 2.5 will review the current legislative efforts to standardize lighting conditions in office rooms, and to regulate the energy consumption of buildings. Section 2.6 will highlight the need for an adaptive daylighting controller because of the statistical nature of most visual discomfort indices. We will discuss in section 2.7 the difficulties in modeling the daylighting illuminance in an office room equipped with venetian blinds, a very common situation in modern buildings. Section 2.8 will review recently proposed advanced daylighting control systems and the philosophies driving them. Finally in section 2.9 we will briefly summarize the requirements of the control algorithm that we are going to develop, implement and test in this work.

2.1 Global warming and carbon emissions

> My entirely unscientific impression of the recent weather is not just that it's getting hotter—it's getting weirder.
>
> *(Slashdot)*

On 7 December 2006 a 20 m wide tornado, rated 1–2 out of 6 on the Fujita scale, damaged 150 homes and hurt 6 people in Kensal Rise, in the north-west of London (see Figures 2.1 and 2.2). Tornadoes by themselves are not rare sights in the UK—the British Isles are hit

Figure 2.1: The Kensal Rise tornado.

Figure 2.2: The damage caused by the Kensal Rise tornado.

by as many as 40 tornadoes each year—but this was the first time since December 1954 that the capital itself had been hit with such damage. Neither was this the first freak tornado to hit the UK in recent years—a year earlier, in July 2005, a tornado rated 3–4 on the Fujita scale, the worst in 25 years, hit Birmingham and caused devastating damage to homes and businesses.

The public's memory has not yet forgotten the extreme weather events of the past five years: the 2002 floods in eastern european countries; the 2003 heat wave, which killed more than 14 000 people in France alone; the 2005 floods in Lucerne, Switzerland; and the 2005 hurricane season, from which New Orleans has not recovered yet. The twentieth century has been the warmest of its millenium, the 1990s have been their century's warmest decade and 1998 has been the warmest year of the millenium in the northern hemisphere. Eleven of the last twelve years rank among the twelve warmest years in the instrumental record of global surface temperatures.

Global warming—the slow but steady increase of Earth's global temperature—is believed today to pose a grave danger to the stability of our environment, if it continues or (as some scenarios suggest) accelerates. Although it is difficult to directly relate global warming with the extreme weather events described above, the evidence suggests that more such events are to be expected if nothing is done to mitigate the climate change.

The global temperature, a weighted average of land, air and sea surface temperatures, has

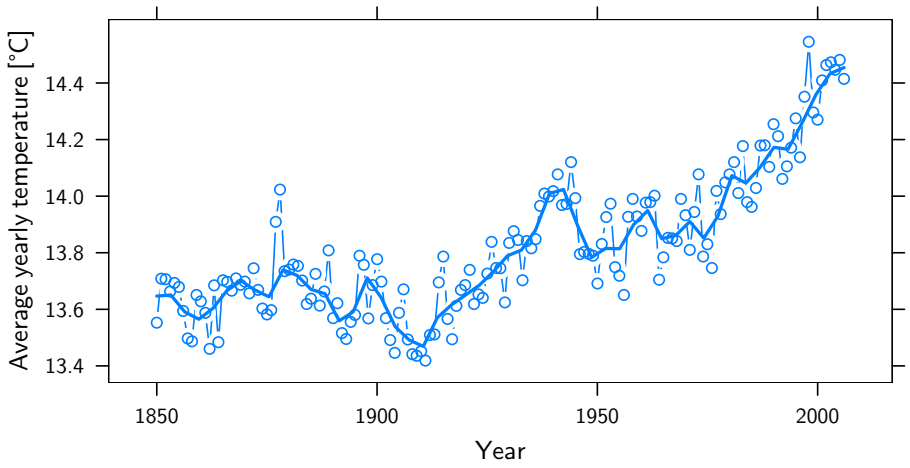


Figure 2.3: Global temperatures 1850–2004, from Jones and Salmon (2006). The solid line is a smoothed curve.

been instrumentally measured since 1850. The data is freely available (Jones and Salmon, 2006) and shown in Figure 2.3. Global temperatures have increased by about 0.8 °C in 150 years. A first increase happened between 1910 and 1945, and a second one from 1975 to this day.

Not all parts of the world have warmed at the same rate. Nor is the temperature increase equally distributed over the whole year. In Switzerland, for example, winters have become about 2 °C warmer during the same time, while summers have remained stable (Bader and Bantle, 2004). This explains why many glaciers in the swiss Alps have either disappeared or been substantially diminished.

The full mechanism responsible for global warming is not yet understood, but there is little doubt that a man-made increase of so-called greenhouse gases contributes to it substantially. The fundamental principles of the greenhouse effect are today very well understood and uncontroversial. We know that several gases can reflect or trap heat from the Earth that would otherwise have radiated out into space.

There is nothing bad about this effect—without it, the Earth would be about 33 °C colder than it is now, making it probably lifeless. But the greenhouse effect becomes a problem when the concentrations of the greenhouse gases—water vapor, carbon dioxide (CO_2), methane (CH_4), nitrous oxide (N_2O), CFC gases and ozone (O_3)—increase beyond their natural levels. The International Panel on Climate Change have recently released their Climate Change 2007 report (Alley et al., 2007), which estimates that mankind's production of carbon dioxide, methane and nitrous oxide has retained globally an extra $2.30 \pm 0.23\,\mathrm{W/m^2}$ of solar radiation between 1750 and 1998, 70% of which is attributable to CO_2 emissions alone (c.f. Figure 2.4).

CO_2 is the main contributor to the anthropogenic greenhouse effect and its global emission is usually taken as proportional to the emission of all other greenhouse gases. Variations in its output are admitted to accompany corresponding variations in the output of all greenhouse gases. Its atmospheric concentration is thus usually taken as a proxy for the total contribution to global warming from greenhouse gases.

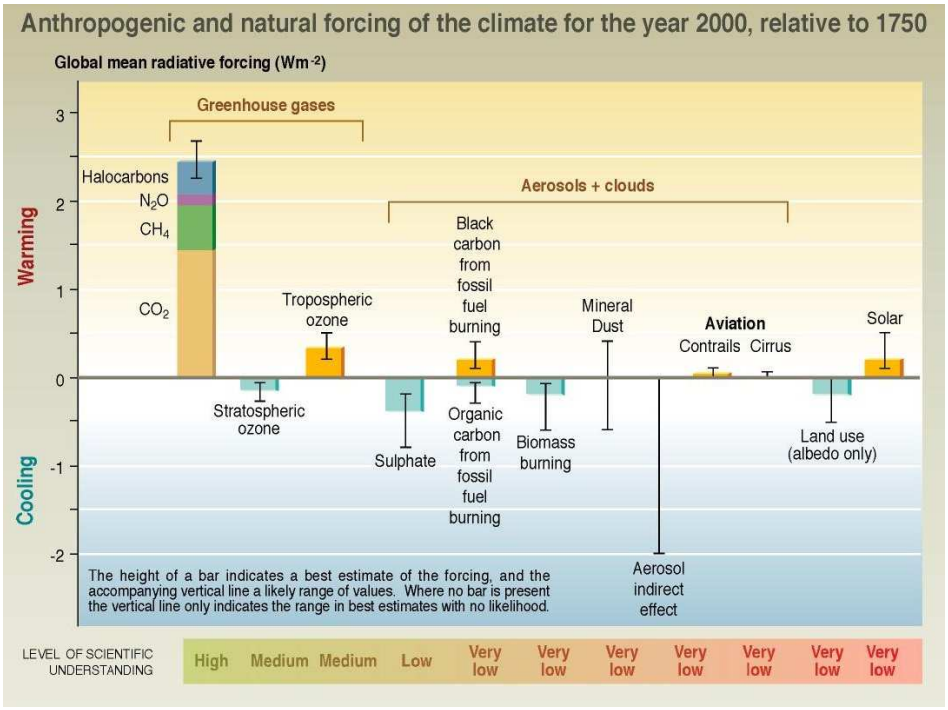


Figure 2.4: Anthropogenic and natural climate forcing, from Alley et al. (2007).

In the 1950s, Professor Roger Revelle, concerned by the global post-World War II economic expansion, was the first to propose a long term research program that would regularly collect samples of CO_2 atmospheric concentrations. He felt it was important to monitor how human activities were influencing the delicate chemical balance of our atmosphere.

Under his direction, Keeling and Whorf began in 1958 a series of monthly measurements of atmospheric CO_2 concentrations on Mauna Loa, about 3400 m above sea level on the barren lava field of an active volcano in Hawaii. The site is considered as one of the best sites for measuring undisturbed CO_2 concentrations because of the complete absence of vegetation or human activities, and influences from volcanic vents can be excluded from the record.

This measurement continued well into the 21st century and is today the longest continuous record of CO_2 concentrations available in the world. The measurements are freely available (Keeling and Whorf, 2005) and are shown in Figure 2.5.

Cleveland (1993) uses this data as an example of an exhaustive graphical data analysis in his classic *Visualizing Data*. The yearly oscillations in CO_2 concentrations are normal and are the signs of a healthy, breathing planet. There is much more landmass in the northern hemisphere, hence more forest, which absorbs the CO_2 during its growing season.

But the main growing trend of about 5% per decade is very worrying. The current CO_2 levels are higher than they have been for 100 000 years (Petit et al., 1999), and there is no doubt that most of this increase results from human activities.

Anthropogenic greenhouse gases are believed to have contributed to most of the observed recent global warming. This view is summarized in the last IPCC report (Alley et al., 2007):

> Most of the observed increase in globally averaged temperatures since the mid-20th century is *very likely* due to the observed increase in anthropogenic greenhouse

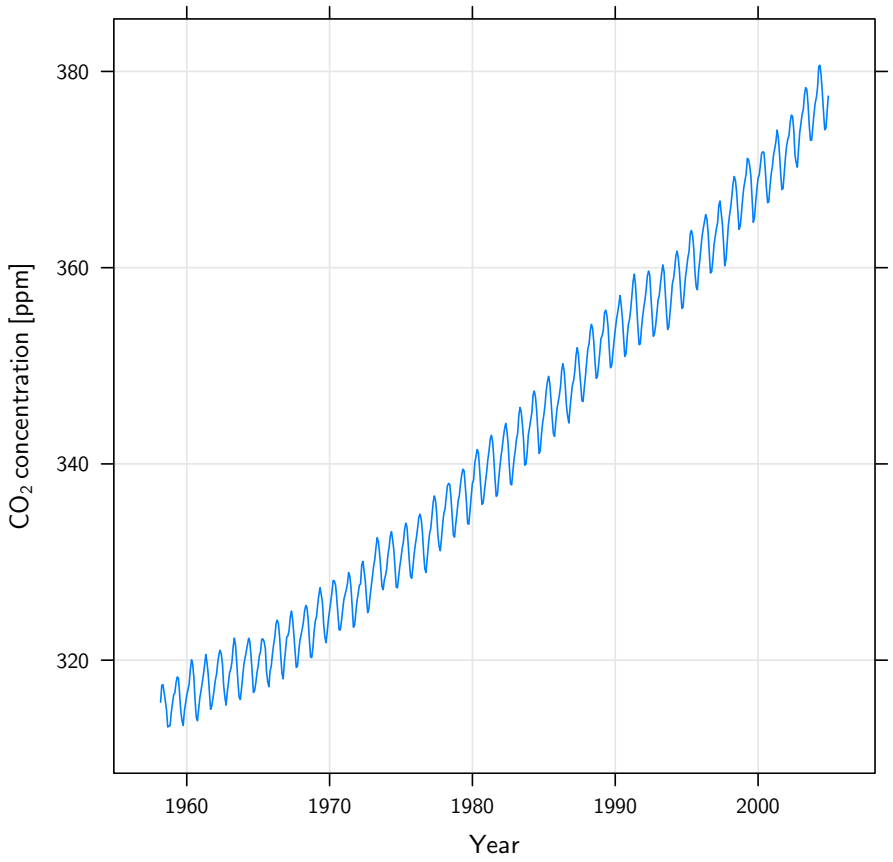


Figure 2.5: Atmospheric CO_2 concentrations, from Keeling and Whorf (2005).

> gas concentrations. (italics in text)

Similarly, a National Academy of Sciences Commitee on the Science of Climate Change report (NAS-2001) finds that:

> The IPCC's conclusion that most of the observed warming of the last 50 years is likely to have been due to the increase in greenhouse gas concentrations accurately reflects the current thinking of the scientific community on this issue.

Oreskes (2004) has conducted a survey of 928 peer-reviewed papers on the topic of "climate change" and found that 75% of these agreed with this consensus view, 25% expressed no opinion, and none disagreed. Neither does Lomborg, author of *The Skeptical Environmentalist* (Lomborg, 2001), disagree with this consensus view, although he is critical of the way the consequences of global warming have been modeled. No peer-reviewed published scientific work disagrees that human activities are the cause of most of the global warming, and we could find no scientific publication that did not agree that if CO_2 concentration were allowed to continue rising, the atmospheric temperatures will eventually rise, causing the polar ice caps to melt, the coastal areas of the continents to flood and the overall global climate to change dramatically.

Skeptics rightfully wonder whether the increase in CO_2 concentrations might not be part of a natural cycle of a shorter period than the geological timescales observed by Petit et al. (1999), but longer than the one for which we have monthly instrumental readings. Robertson et al. (2001) provides such measurments from the analysis of CO_2 concentrations in air bubbles trapped in ice cores. The data is publicly available and plotted in Figure 2.6, which shows how atmospheric concentrations of CO_2 have evolved in the last 500 years. It has been stable between 1500 until the beginning of the industrial revolution around 1850. A drop is suggested between 1600 and 1750, which would coincide suspiciously with the Little Ice Age. But since 1850 the increase in CO_2 concentrations has exploded, reaching levels that have not been seen in 500 years, nor indeed in 100 000 years. The current rise in CO_2 concentration levels thus coincides with the explosion of industrial activity (and CO_2 emissions) that started with the industrial revolution.

Switzerland alone emitted, in 2004, 44.55 millions of tons of CO_2, or almost 6 tons per capita. At standard pressure and temperature, this is more than 3 million liters per inhabitant, the approximate volume of a typical hot-air balloon[1].

Such compelling evidence, and the most elementary prudence, demands that we seek and implement solutions to limit and decrease our emission of greenhouse gases. Broadly speaking, there are two ways to accomplish this. The first, most comfortable, and most tempting solution, is to look for alternative, cleaner energy sources that do not emit greenhouse gases. This solution will, however, treat only the current symptoms instead of curing our disease. Our fundamental problem is not our energy sources—it is the use we make of them. We are addicted to energy, not oil. Human activities have emitted greenhouse gases since the dawn of mankind, and plentiful oil has discouraged any rational use of that energy. Switching to a carbon-neutral, plentiful, sustainable energy source is a laudable goal but not the best one.

The second, and, we believe, best solution, is to rationalize our current energy consumption. We can try to limit our energy consumption, or reduce it by making it more efficient. This is one of the driving ideas behind this work.

[1] The molar mass of CO_2 is 44 g/mol. A perfect gas's volume is 22.4 L/mol. Switzerland has 7.5 million inhabitants. The volume of a typical hot-air balloon is 2500 m^3.

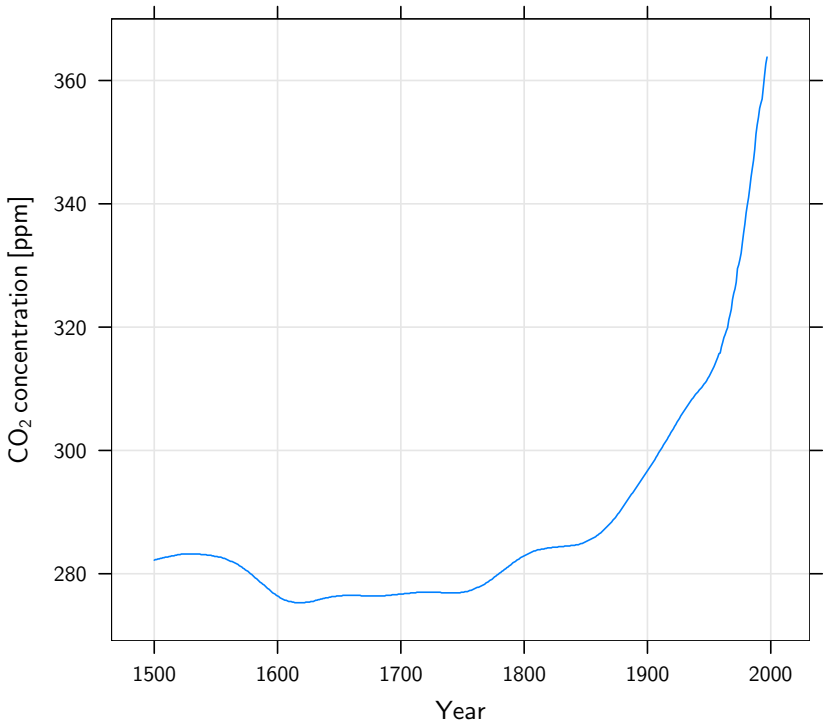


Figure 2.6: CO_2 concentrations for the last 500 years, from Robertson et al. (2001)

According to the International Energy Agency (IEA-2007), the energy demand in 2004 of the "Residential" and "Commercial and Public Services" sectors of the OECD european countries was 461 984 thousands of tonnes of oil equivalent (ktoe), out of a total of 1 333 497 ktoe[2]. These sectors use therefore about 35% of our energy (against 23% in 1991, c.f. Schipper et al. 1996), most of which is used for heating, cooling and lighting our buildings. Countries whose energy supply depends on fossil fuels should therefore reduce the energy demands of their buildings not only for economic reasons, but for environmental reasons too[3].

The energy demands of buildings can be reduced through many different technologies. Inexpensive measures include weatherstripping, caulking, and insulating walls, floors and ceilings. According to Cunningham and Saigo (1995), prototype buildings have achieved up to 94% energy savings compared to the market average, and 83% compared to the most efficient building on the market. Zero-energy buildings, i.e. buildings that produce more energy than they use, are today technologically possible—the Pearl River Tower, currently under construction in Guangzhou in China, will be one of the world's first zero-energy skyscrapers when it is finished in 2009 (WIKI-PRT). In this work, we will not deal with changes to a building's infrastructure but instead focus on how a building's operation can be improved to make more efficient use of daylight.

According to the *IESNA Lighting Handbook* (Rea, 2000, p. 26-1), 20–25% of all electricity in US buildings, or 5% of the national energy consumption, is used for lighting. The heat generated by this same lighting represents between 15–20% of a building's cooling load. It makes therefore sense to develop control algorithms for building management systems that minimize the use of energy for lighting.

The *Handbook* quotes field studies (Rea, 1984a) according to which 40% energy savings are possible with elementary control systems, such as predictable scheduling where lighting elements are connected to timers. Unpredictable scheduling (i.e., that relies on occupancy sensors) has achieved up to 60% savings in some areas, and in extreme cases (Rubinstein et al., 1984) up to 80% savings on lighting energy.

The dynamic range of daylight is about five times that of electric lighting, and architects and contractors have extensively used daylight since almost two decades to replace or supplement electric lighting. Daylight not only provides illuminance to the building occupants instead of electric lighting, but an optimal usage of the building's solar shading devices can help reduce the energy demand for heating in winter and the cooling load in summer. Studies (Guillemin, 2003) have shown that state-of-the-art integrated control algorithms can reduce the overall energy consumption (lighting and heating) of a non-residential building by about 25%. Nevertheless, the instantaneous daylighting illuminance can be more than twice or less than half the mean design values according to Rea (2000), so great care must be taken when designing a blinds control system.

[2]The world consumes today about one cubic mile of oil (CMO) yearly, and Goldstein and Sweet (2007) argue we should normalize all energy units to that quantity. The density of oil is about 920 kg/m^3 and one cubic mile is 4.17 cubic kilometers, so we would therefore say that the OECD countries use 0.12 CMO per year on these sectors, out of 0.35 CMO. 0.12 CMO is enough oil to cover Switzerland with a 1.2 cm thick layer of oil.

[3]Switzerland's electricity, about 23% of that country's final energy use, is almost entirely produced by carbon-neutral dams and nuclear plants. But electricity powers the cooling and lighting of most buildings, whereas their heating is powered by fossil fuels. Switzerland should therefore concentrate on achieving savings on heating energy, rather than on cooling or lighting.

2.2 Inadequacies of manual blinds control

> That which is common to the greatest number has the least care bestowed upon it. Every one thinks chiefly of his own, hardly at all of the common interest; and only when he is himself concerned as an individual.
>
> *(Aristotle)*

If an intensive use of daylight can help reduce the energy bill of the buildings sector and provide a pleasant environment to the occupants, why not let the occupants themselves manage their shading devices?

The problem with completely manual blinds and electric lighting controls is that we humans are fundamentally lazy. Or to put it in a less unfavorable way: we don't mind small levels of discomfort, especially if the alternative is to continuously adjust a shading device.

To ask building occupants to manage their shading devices in a continuous, optimal way is unrealistic. But one might hope, at least, that the occupants' behaviour towards their blinds show some degree of rationality and that energy savings can still be obtained. This is, unfortunately, not the case for the majority of building users, as has been reported in the scientific literature.

Sutter et al. (2006) have recently monitored the use of venetian blinds in eight offices over 30 weeks, measuring the settings of the blinds every 15 min. Their data helped them validate or invalidate certain hypotheses on the manual use of shading devices. From their study, we should note the following:

- The use of shading devices is consistent, i.e. similar conditions cause similar usage patterns.
- The use of shading devices depends on how easily accessible the controls are, and their type (manual or motorized). Motorized venetian blinds were used three times more often than manual fabric blinds. A previous study had found that manual fabric blinds were used as little as once per month(!) (Paule, 2006).
- Most of the time, the venetian blinds are either fully retracted or fully closed. Whether this is influenced by the type or placement of the blinds' controls is not discussed by the authors. In our building, once motorized blinds start moving, they will not stop until the user presses the button again.
- There is hysteresis in the use of the blinds. The window luminance, or the indoor illuminance, at which the users raise their blinds are not the same at which they lower them again.

Although the manner in which the users use their blinds is consistent, these points suggest that it is not optimal. From everyday's experience we know that nobody will adjust their blinds regularly but will only do so once a certain threshold of discomfort is reached. The position of this threshold might even depend on the type and placement of the blinds' controls. In addition, few people consciously adjust their blinds before leaving their office to optimize the use or rejection of solar gains. This has been confirmed by Galasiu and Veitch (2006) in a literature review, in which they found that people tend to set their shading devices in

a conscious and consistent way—though not necessarily rational—and to forget about them. "Conscious", in this sense, means the users know why they are adjusting their blinds, while "consistent" means that similar external stimuli will yield similar manual blinds' settings. This was already pointed out by Rea (1984b). Indeed Galasiu and Veitch recommend that more research projects be carried out to help us understand exactly how the user's preferences are distributed as a function of external stimuli.

2.3 Health benefits of daylight

> Daylight makes the hills and valleys stand out like the folds of a garment, clear as the imprint of a seal on clay.
>
> *(Job 38:14)*

Solar radiation can warm buildings in winter and save back-up energy, but has also recently been shown to have measurable health benefits.

Retinal ganglion cells were discovered in 2002, a previously unknown connection between the eye and the circadian pacemaker in the mammal brain that drives daily wake-sleep cycles and certian hormonal levels (Berson et al., 2002).

Webb (2006) reviewed recently the non-visual effects of light on the human body, and describes how the current bias towards visible light might affect us physiologically. Blue light is known to affect the circadian rythm, mood and even behaviour. Skin exposure to ultraviolet light causes vitamin D synthesis, and exposure to strong white light is regularly prescribed to treat Seasonal Affective Disorder, whose sufferers experience depressive symptoms in winter.

In Galasiu and Veitch (2006) we read that there is evidence from Begemann et al. (1997) that the physiological need for lighting might even vary over the course of the day, in response to the circadian rythm. They note also that keeping a constant horizontal workplane illuminance might not be optimal. Recent research suggests that human bodies need a lighting environment as close as possible to the natural daily cycle.

These researchers, and many others, suggest that any building construction code or building lighting design based solely on a specification of maintained illuminances fails to satisfy the spectral requirements of the human body. The development of a daylighting control system that would take these needs into account is a research project in its own right, but the present work was started before these effects were understood and our controller will not use these findings.

2.4 Difficulties in defining visual comfort

> One measures a circle beginning anywhere.
>
> *(Charles Fort)*

2.4.1 Early ergonomical studies

Many disciplines that we call sciences in modern times started out as arts or crafts, i.e. as sets of rules or general know-how that was known to work without really understanding how or why. Shipbuilding, for instance, was until a few centuries ago not an exact science but relied

entirely on the shipbuilder's experience and prior work—or lack thereof. The *Vasa* swedish ship of the line was the most powerful ship of the world when launched from Stockholm on 10 August 1628—until it capsized and sank, less than 1000 m into her maiden voyage. Its complete design was in the head of the shipwright, based on shifting specifications given by the King. Now that shipbuilding has become a science, we know that if the ship's center of gravity had been about 10 cm lower it would have remained afloat.

Almost no discipline related to human welfare can today be called an exact science. Medical science is probably the most scientific-like of such disciplines, having also evolved from empirical roots, but remains even today a science that relies heavily on probabilities and statistics.

Our eyes are not optimally adapted to life indoors, to its short distances, and in particular to its relatively low illuminances. The lighting in most of our modern workplaces, be they factories, hospitals, offices, shops or workshops, is unnatural and unsuitable for our natural condition. Understanding what makes an environment visually pleasant and healthy must therefore be regarded as a difficult, inexact discipline that has immensely benefited from science but that is still far from being completely consolidated.

There is no formal, universally accepted way of quantifying visual comfort. If there was one, whose inputs were readily measurable quantities, a daylighting controller could be built provided one knew how to control those quantities. We will discuss the most popular proposals currently used in buildings codes in section 2.4.3 but first we will review early, more empirical work on this matter.

Luckiesh and Moss (1937) is one of the earliest works on the topic of visual comfort still easily available. Their text summarizes research that was carried out during and after World War I, when it was important to maximize the industrial output from factory workers. The effect of lighting on productivity was one such parameter that was investigated during that time.

The authors note that to establish a suitable visual environment one must ask oneself three questions: 1) What kind of light do we have? 2) How much of it do we have? and 3) What do we do with it? The first two questions deal with what kind of *Light* is available, while the third one deals with what *Lighting* we are to produce.

Nowadays, the production of light of any quality and quantity is no longer an economic problem. Until 1880, mankind had to carry out most of its indoor activities with one-candela light sources, under conditions that would be considered intolerable today[4], whereas the cost of the lighting equipment in a modern building is a small fraction of the overall building's cost.

The challenges left to the designer of a lighting control system are therefore to provide a visual environment that makes human activities possible and comfortable with whatever lighting is available, while minimizing its energy consumption. In this work we will develop a controller for a pre-existing installation and there will be no guarantee that the office could not be lit in a better way, but the visual environment we provide will be the one that makes the optimal compromise between comfort and energy with the current lighting installation.

A common metric used in lighting prescriptions is the illuminance of the surface where most of the work is being done. If this task illuminance is constant across the surface, it

[4]But which unfortunately are still the norm in many countries. Luxtreks (`http://www.luxtreks.com`) is a not-for-profit organisation dedicated to donating solar-powered, battery-equipped lighting units to remote villages in places such as Bolivia, Peru or Tanzania. Their beneficiaries suffer often from lung damages caused by toxic fumes emitted from their crude *ghee* lamps (burning clarified butter) or kerosene lamps.

must fall between two extremes: 1) an illuminance just sufficient for discerning the details of the work at hand, for the duration of the work; and 2) an illuminance that provides the easiest perception of the work surface. Any illuminance below this minimum will make the work impossible while any illuminance above this maximum is a waste of energy (and a risk of glare).

We might be tempted to search for such an intermediate illuminance, balancing the energetic cost with the economic cost of having people not work in the easiest and most productive conditions, but this is impossible in practice. Instead, as we will see, our system will try to find a *practical* illuminance (retaining the terminology introduced by Luckiesh and Moss), that is, an illuminance that still permits productive and healthy work to be done while keeping energy expenditures low.

It is, of course, slightly ironic to spend so much energy on achieving a given illuminance, which is something the eye is totally insensitive to. The *brightness*, or luminance of a surface, is what is directly perceived by the eye, and depends not only on the surface's illuminance but also on its reflectance and specularity. For the relatively diffuse, homogenous surfaces usually encountered in the workspace we will assume that the luminance of a surface as seen by the observer is proportional to its illuminance.

Luckiesh and Moss give one of the earliest tables of recommended illuminances for different tasks. They recommend 20–50 footcandles (fc), or 215–540 lx, for typical office work, i.e. "moderately critical and prolonged tasks, such as clerical work, ordinary reading [...]".

They also relate an experiment whereby 82 schoolteachers were asked to adjust the illuminance they deemed necessary for reading black print on white paper for extended periods of time. The results are shown in Figure 2.7. This experiment illustrates the extreme variations that can be found from individual to individual (a factor of 100 more illuminance in the extreme cases) and suggests that global illuminance prescriptions are not sufficient, but that illuminances should be adjusted for (or by) each individual.

Luckiesh and Moss deal with adequate illuminances but provide little guidance on the avoidance of excessive lighting levels. At the time their book was written, daylight was not yet widely incorporated in building design, nor were electric lighting fixtures strong enough to cause serious glare problems.

Weston (1935, 1945) has studied the relationship between task performance and task illuminance, especially above the visibility threshold. He has shown that in general, task performance increased first rapidly with task illuminance until a point is reached where large changes in task illuminance have only small effects. It is tempting to conclude that there is no reason to provide more task illuminance than that required for an efficient execution of the task—but a task illuminated just enough to carry it out cannot be sustained for prolonged periods of time. There is more to visual comfort than just the speedy execution of a short task.

Etienne Grandjean has carried out important research in every aspect of comfort in the work environment, including visual comfort in the computerized environment. In *Ergonomics in Computerized Offices* (Grandjean, 1987) he defines glare as "a gross overloading of the adaptation processes of the eye, brought about by overexposure of the retina to light." He further distinguishes between "relative glare", caused by "excessive brightness contrasts between different parts of the visual field"; "absolute glare", caused by sources so bright that the eye cannot physiologically adapt to them (e.g. the sun); and "adaptive glare", a temporary effect experienced, for instance, when coming out of a dark room into bright daylight.

He recommends that "All important surfaces withing the visual field should be of the same

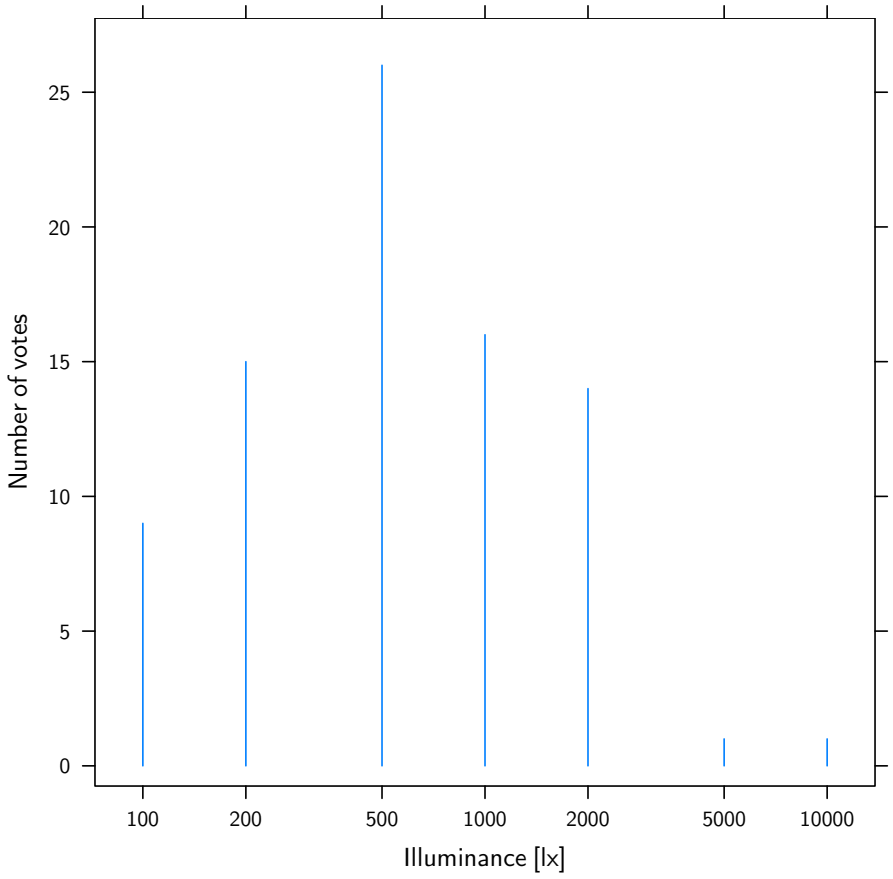


Figure 2.7: Preferred illuminance among 82 schoolteachers, from Luckiesh and Moss (1937). The horizontal scale is logarithmic.

order of brightness", and that "The general level of illumination should not fluctuate rapidly because pupil reaction as well as retinal adaptation is a relatively slow process." His work is one of the earliest works recommending spatial and temporal uniformity of luminances in the field of view, in addition to a suitable range of illuminance and avoidance of glare.

He cites a study of 15 open-plan offices and 519 employees, where a workplane illuminance of 1000 lx or more resulted in a statistically significant increase in reported eye complaints. Employees preferred illuminance ranging between 400–850 lx. He points out that it is certainly not the workplane illuminance itself which was the source of glare. A brightly lit office room is rather more likely to have problems with reflections, deep shadows and relative glare.

Experiments showing preferred illuminances between 1000–4000 lx should, according to Grandjean, be eyed with suspicion. Such findings could be artefacts that result from uncarefully designed brightly lit backgrounds.

Grandjean recommends that office jobs without visual display terminals (VDT) should be illuminated by 500–700 lx, with brighter values for elder people. VDT tasks where most of the time is spent staring at the screen should be given 300 lx. These recommendations, however, were given at a time when computer screens were black with monochrome display and whose letters had luminances of 40–50 cd/m^2. It is not clear whether these prescriptions are still valid today.

Refering to general lighting design and placement of luminaires, Grandjean gives the rules summarized in Table 2.4.1.

Even today, reflections on computer screens are a major cause of discomfort. These should be avoided, and the single most effective measure one can take according to Grandjean is to adequately position the screen with respect to lights, windows and other bright surfaces.

He concludes by recommending between 300–500 lx for conversational tasks, i.e. tasks during which one often glances at the screen, and 500–700 lx otherwise. Again, these values should probably be revised since the widespread introduction of high-luminance flat screens.

Grandjean is also the author of *Fitting the Task to the Man* (Grandjean, 1988), whose chapters 17 (Vision) and 18 (Ergonomic principles of lighting) are of interest to us. These chapters pick up where his earlier book left off and complement it with more definitions and prescriptions.

He defines "visual acuity" as the "ability to perceive two lines or points with minimal intervals as distinct". Visual acuity is essentially what enables us to carry out our work efficiently and comfortably. It varies as follows:

1. It increases with the illuminance, plateauing at illuminances above 100 lx.
2. It increases with the contrast between the test symbols (letters on paper or on a screen) and their immediate background.
3. It is greater for dark symbols on a bright background than the reverse.
4. It decreases with age, down to about 50% at 80 years old.

Since visual acuity does not greatly improve with higher illuminances, Grandjean cautions again against more than 1000 lx in office spaces and recommends task illuminances in the 500–700 lx range.

Grandjean points out that the German (DIN) and American (IES, 5th ed, 1972) requirements for the same tasks are significantly different. The American values are systematically

Lighting design and luminaire placement	Daylighting design
All large objects and major surfaces in the visual environment should, if possible, be equally bright.	High windows are more effective than broad ones, since the light penetrates further into the room. The lintel should not be deeper than 300 mm.
The line from eye to light source must have an angle of more than 30° to the horizontal plane. If a smaller angle cannot be avoided, then the lamps must be shaded more effectively.	Window sills should be at table height. If the window extends below the table-top it will be cold in winter and may cause glare.
The working area should be brighter in the middle and darker in the surrounding field.	The distance from window to workplace should not be more than twice the height of the window.
To avoid annoying reflections from the desk surface, the line from eye to desk should not coincide with the line of reflected light.	For workrooms the window area should be about one-fifth of the floor area. This is only a general rule which is very flexible according to circumstances.
Surfaces in the middle of the visual fiels should not have a brightness contrast of more than 3:1.	It is important that the glass should transmit plenty of light flux. Clear glass has a transparency of more than 90%, whereas frosted glass, glass bricks, or special heat-insulating glass may have transparencies from 70% down to only 30%.
Contrasts between the central and the marginal areas of the visual field should not exceed 10:1.	Effective protections against the glare of direct sunlight, and against radiant heat, are important in securing good visibility and comfort indoors. The most efficient method is an adjustable external sunshade, either venetian blinds or shutters. Venetian blinds inside the window, or between the panes of double-glazing, are a mistake, because they afford no protection against radiant heat.
Light sources should not contrast with their background by more than 20:1.	Each window should receive direct light from the sky vault, and it is desirable that a portion of sky should be visible from every workplace.
The maximum brightness contrast within the entire room should not exceed 40:1.	The nearest building should be at least twice as far away as its own height.
No light source should appear within the visual field of an office employee during the working activities.	Pale colours should be used, both in the room itself, and in any courtyard outside, so as to reflect as much of the incident daylight as possible.
All lights should be provided with shades or glare shields to prevent the luminance of the light source exceeding 200 cd/m^2.	
Fluorescent tubes should be aligned at right angles to the line of sight.	
It is better to use more lamps, each of lower power, than a few high-powered lamps.	
Excessive contrasts are more troublesome at the sides than at the top of the visual field.	
The use of reflecting colours and materials on table-tops or office machines should be avoided.	

Table 2.1: Lighting design, luminaire placement, and daylighting design rules, from Grandjean (1987, 1988).

higher. The recommended illuminance for office work are a mere 500 lx in Germany but 1600 lx in the US. This should be taken as anecdotal evidence that official lighting recommendations do not depend only on genuine ergonomic principles, but also on political and economical ones, such as the price of energy.

Fitting the Task to the Man concludes with design rules with respect to daylight, that we have also summarized in Table 2.4.1.

2.4.2 Preferred range of illuminance

> It should be kept in mind that lighting is not an exact science. It deals with people as well as things, and the lighting in a given interior is not good unless the occupants like it. An awareness of the fact that lighting is as much an art as a science is, indeed, central to a full appreciation of what is important in interior lighting.
>
> *(CIE Guide on Interior Lighting, second edition)*

In addition to the experiments described in the previous section, other researchers have conducted similar experiments where volunteers were asked to rate illuminances, in order to estimate what the optimal range of illuminance for typical office work should be.

Fischer (1970) has pooled the results of such experiments conducted by Balder (1957), Muck and Bodmann (1961), Söllner (1966), Riemenschneider (1967), Westhoff and Horeman (1963), Boyce (1968), and Bodmann et al. (1963). In each experiment the distribution of the reported preferred illuminances on a logarithmic scale have been fitted with a normal distribution. From these plots we have estimated the full width at half-maximum of each distribution and deduced the fitted standard deviation.

The *IESNA Lighting Handbook* (Rea, 2000, page 3-39) gives some additional illuminance ranges from Bodmann (1962), Saunders (1969), Bean and Hopkins (1980) and Nemecek and Grandjean (1973), without specifying if these are confidence intervals, a fitted or sample standard deviation or the standard error on the mean estimate. For example, Nemecek and Grandjean report that:

> [...] the frequency of eye troubles in the three offices with more than 1000 lx was significantly higher than in the other ones ($p < 0.001$). Another analysis revealed that lighting intensities between 400–850 lx were judged to be the best.

whereas Saunders report that:

> [...] at 400 lx, approximately 85% of the observers found the lighting conditions satisfactory or better, the figure increasing to 95% at 1000 lx.

Figure 2.8 summarizes these findings, assuming the work quoted by the *IESNA Lighting Handbook* refers to the fitted standard deviation. The study due to Vine et al. (1998) was included in a similar fashion.

Nabil and Mardaljevic (2005) have carried out one of the most recent reviews of the current understanding of preferred illuminances. They conclude that:

- Daylight illuminances lower than 100 lx are generally considered insufficient to be either the sole source of illumination or to contribute significantly to artificial lighting.

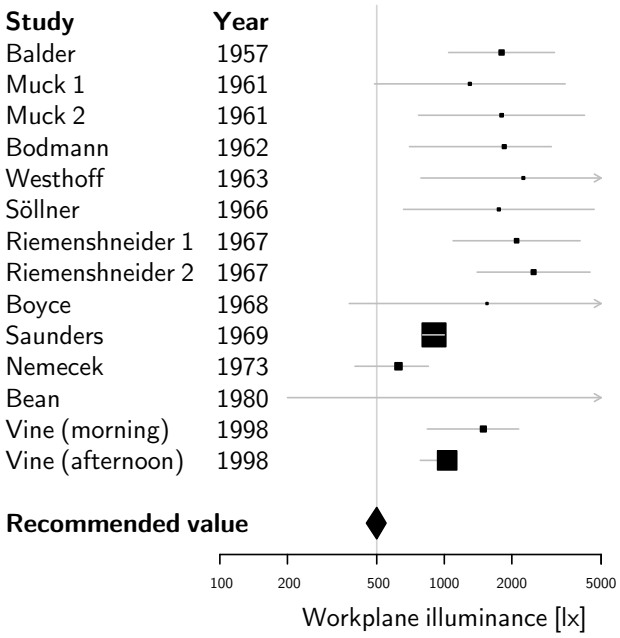


Figure 2.8: Preferred illuminances in office rooms, pooled data. The width of each line is the fitted standard deviation, with some exceptions (see text). The size of each dot is inversely proportional to the fitted standard deviation. The vertical line is the recommended value of 500 lx one often finds in lighting design codes.

- Daylight illuminances in the range of 100–500 lx are considered effective either as the sole source of illumination or in conjunction with artificial lighting.
- Daylight illuminances in the range of 500–2000 lx are often perceived either as desirable or at least tolerable.
- Daylight illuminances higher than 2000 lx are likely to produce visual or thermal discomfort, or both.

From these observations they even propose a new measure of a building's daylighting performance, christened "Useful Daylight Illuminance" (UDI), defined as the total time in the year the workplane's illuminance is between 100–2000 lx. This range of illuminances was chosen precisely as a result of this literature review. They propose that this metric shall replace the Daylight Factor as an indicator of daylighting performance because:

> It is recognized that the daylight factor approach offers only a limited insight into true daylighting performance because it is founded on a measure of illumination under a single, idealized overcast sky.

Figures 2.7 and 2.8 should make it clear that the range of preferred illuminances can vary greatly from individual to individual, a phenomenon recognized by Galasiu and Veitch (2006). Even within the restricted range of 100–2000 lx proposed by Nabil and Mardaljevic there is plenty of manoeuvering room for a lighting controller to achieve energy savings, instead of imposing a constant task illuminance on all occupants.

2.4.3 Objective quantification of visual discomfort and glare

> If there is a universal mind, must it be sane?
>
> *(Charles Fort)*

Throughout this work we will place much emphasis on the need to quantify objectively the visual discomfort. This notion might, at first, sound very odd. We are all very capable of determining by ourselves whether a visual environment is pleasant and productive or not, and nobody has ever walked around a building with an instrument measuring the visual discomfort (even though modern technology would in principle permit it). However, a rational assessment of visual discomfort is, in our opinion, important for three reasons.

First, without such a tool, the building designers must rely on their expertise and rules of thumb to ensure that a planned building will satisfy the visual requirements of its future occupants. More often than not, these designers aim to satisfy national building construction codes that might fail to account for some peculiarity of the building, ruining in some cases the visual comfort.

Second, daylight plays an important role in newly designed buildings because of the potential energy savings. But the use of daylight poses glare problems of its own that can be overlooked unless one quantifies objectively the visual discomfort.

Third, the performance of most daylight-responsive control algorithms (including the present one) will benefit from a rational quantification of the visual discomfort. Coupled with an accurate daylighting model, the controller's algorithm is free to explore its degrees of freedom and find the combination of blinds' settings and electric lighting power that provides an optimal visual environment.

In this section we will review the current understanding of this problem and the recommendations used by practitioners.

Glare and insufficient illuminance are two main causes of visual discomfort in interior environments, but are usually treated separately in the literature and in building codes. Glare is, by far, the most difficult problem of the two.

Early work by Guth and Hopkinson focused on finding a mathematical relationship between glare perception and the distribution, size and intensity of light sources. Field studies led to the determination of Guth's Discomfort Glare Rating and to Hopkinsons's Glare Index in the early sixties.

The Commission Internationale de l'Eclairage (CIE) compiled these results in 1983 and published a report *Discomfort Glare in the Interior Working Environment* (CIE, 1983) on the state-of-the-art on discomfort glare[5] in the interior working environment. In the same report, the CIE recommends the adoption of a formula proposed by Einhorn, considered as the best compromise between different national systems. This formula led to the CIE Glare Index (CGI), defined by:

$$\mathrm{CGI} = 8 \log_{10} \left[2 \cdot \frac{1 + E_\mathrm{d}/500}{E_\mathrm{d} + E_\mathrm{i}} \sum_s \frac{L_s^2 \omega_s}{p_s^2} \right] \tag{2.1}$$

where E_d is the direct vertical illuminance [lx] at eye level from all sources, E_i is the eye-level indirect (excluding the glare source) illuminance [lx], L_s is the luminance [cd/m^2] of the bright part of each source s in the direction of the eye, and ω_s is the solid angle [sr] of the latter. p_s is an index proposed by Guth that gives different weights to luminous sources according to their position in the visual field: sources close to the center of the field of view will carry more weight than sources in the field of view's periphery. This index is defined by:

$$\begin{aligned} p_s \;=\; & \exp\left[(35.2 - 0.31889 \cdot \alpha - 1.22 \cdot \exp(-2\alpha/9)) \cdot 10^{-3} \cdot \beta \right. \\ & \left. + (21 + 0.26667 \cdot \alpha - 0.002963 \cdot \alpha^2) \cdot 10^{-5} \cdot \beta^2 \right] \end{aligned} \tag{2.2}$$

where α is the angle [rad] from vertical of the plane containing the source and the line of sight and β is the angle [rad] between the line of sight and the line from the observer to the source. This function is plotted in Figure 2.9.

The CGI made it difficult, however, for luminaire manufacturers to provide design aids such as luminaire data sheets. This, and changes in the working environment, led in 1995 to the publication of another CIE report *Discomfort Glare in Interior Lighting* (CIE, 1995) in which the Unified Glare Rating (UGR) was introduced, defined by:

$$\mathrm{UGR} = 8 \log_{10} \left[\frac{0.25}{L_\mathrm{b}} \sum_s \frac{L_s^2 \omega_s}{p_s^2} \right] \tag{2.3}$$

where $L_\mathrm{b} = E_\mathrm{i}/\pi$ is the background luminance [cd/m^2] seen by the eye of the observer. The UGR is thus a simplified version of the CGI in which no E_d appears, because "for the simplified glare calculation procedures [...] it has not been possible to find a way to include the direct illuminance."

[5] Discomfort glare is defined as glare that causes discomfort without necessarily impairing the vision of objects. It is distinct from disability glare, which is defined as glare that impairs the vision of objects without necessarily causing discomfort.

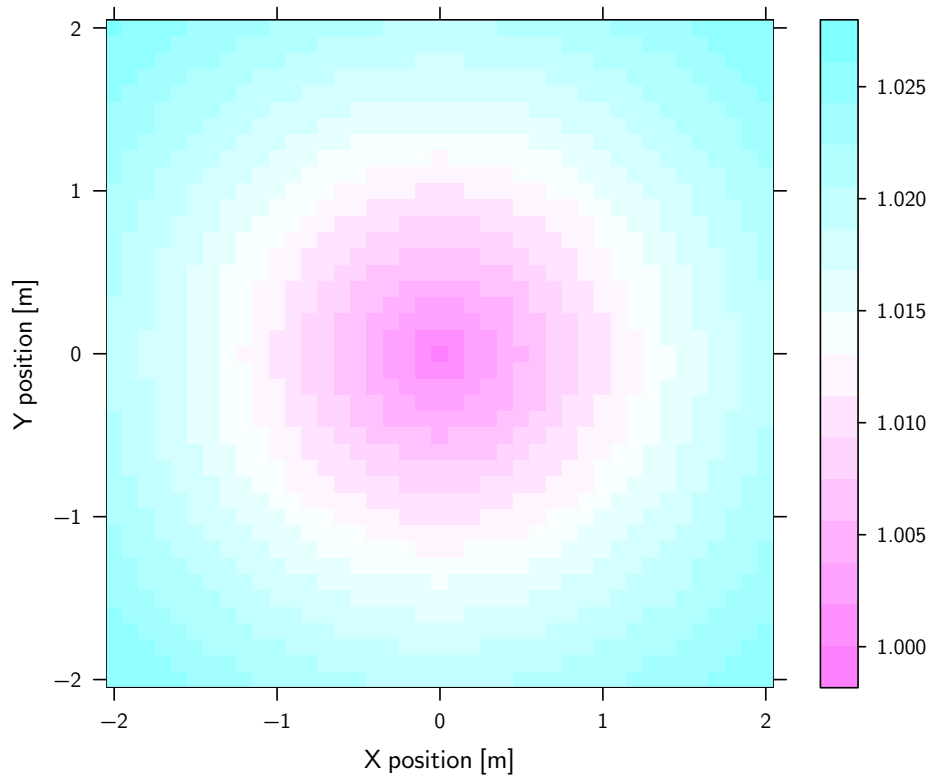


Figure 2.9: Guth's position index p_s for a source on a plane 3 m away and perpendicular to the line of sight.

VCP	50%	60%	70%	80%	90%
UGR	24.0	21.6	19.0	16.0	11.6

Table 2.2: Correspondence between the UGR and VCP.

The UGR takes on typically values between 10 and 30. One glare rating unit is the least detectable step, while three glare rating units are considered the finest granularity that makes sense for normative purposes. UGR = 14 and lower corresponds to the least perceptible glare effect. The standard recommends the UGR be rounded to a value belonging to 13–16–19–22–25–28. The advantage of this new formula was its relative simplicity, which would help luminaire designers provide design aids and simplified glare calculation procedures. Today, however, both indices are readily available from computer simulations.

The UGR is very popular among european lighting designers. American designers prefer the Visual Comfort Probability (VCP), whose expression is given in the *IESNA Lighting Handbook* (Rea, 2000) but is too involved to be reproduced here. The *Handbook* notes the correlation between this index and the UGR and recommends a table of correspondence between the two, given in Table 2.2.

Neither the CGI nor the UGR has been validated against daylight glare or against large light sources. Glare from windows has been simulated in experiments at the Building Research Station in England and at the Cornell University in the USA, using fluorescent lamps behind diffusing screens. From these experiments, the "Cornell formula" (Chauvel et al., 1982) or Daylight Glare Index (DGI) has been derived, defined by:

$$\mathrm{DGI} = 10 \log_{10} 0.48 \sum_s \frac{L_s^{1.6} \Omega_s^{0.8}}{L_\mathrm{b} + 0.07 \omega_s^{0.5} L_s} \tag{2.4}$$

where L_s is the source luminance [cd/m^2], L_b is the background luminance [cd/m^2], ω_s is the solid angle of the source [sr], and Ω_s is the solid angle [sr] of the source modified by its position in the field of view of the user.

Even the DGI has been derived from electric lighting though, and Chauvel et al. (1982) found that a direct application of the DGI formula to a real daylit window overestimates the sensation of glare. For example, a daylit window that would in theory have a DGI rating of 16 is perceived to be about as glary as a luminaire whose DGI would be 10. Recent research by Tuaycharoen and Tregenza (2005) suggests that people are more tolerant towards visually stimulating sources of glare that carry some form of information, instead of plain white lights[6].

The CIE, in its report *Collection on Glare* (CIE-2002), points out that the UGR yields glare estimates that are too severe for small ($<0.005\,\mathrm{m}^2$) sources and too weak for large ones ($>1.5\,\mathrm{m}^2$). For example, a straightforward application of the UGR for a 15 W incandescent lamp, 2 m above eye level, 4 m away, in $30\,\mathrm{cd/m}^2$ background luminance yields an UGR of 39, i.e. totally intolerable glare.

According to this report, research has shown that for small sources in interior lighting it is the *projected area of the source*[7], rather than its solid angle (which for small sources is the projected area divided by the distance), which should be used in the calculation of its

[6] Which does not necessarily mean that the visual comfort is improved. The sensation of glare can be reduced without necessarily improving the visual comfort, in the traditional sense of "capacity to carry out some work efficiently".

[7] I.e., the area obtained by projecting the source on a plane whose normal is the line of sight.

Illuminance [lx]	300	600	1000	1600
UGR	13	16	19	22

Table 2.3: UGR for large glare sources.

luminance. All small sources are therefore taken as being perceived as of size $A = 0.005\,\mathrm{m}^2$, or as discs 80 mm in diameter, no matter their distance to the observer. Therefore, their luminance L [cd/m^2] shall be given as:

$$L = \frac{I}{A} = 200 \cdot I \tag{2.5}$$

where I is the intensity [cd] of the source.

The summand in the UGR formula should therefore be replaced with:

$$\frac{L_s^2 \omega_s}{p_s^2} = 200 \frac{I_s^2}{r_s^2 p_s^2} \tag{2.6}$$

where r_s is the distance [m] to the source. This correction is valid for sources more than 5° off the line of sight at typical indoor lighting distances.

For large glare sources, i.e. luminous ceilings, the standard does not recommend any glare formula at all but gives a table of illuminances the ceiling might provide together with their corresponding UGR rating. These values are given in Table 2.3.

The report proposes a new glare index called the GGR, that redefines the UGR in order to make the transition between "normal" sources and large sources continuous:

$$\mathrm{GGR} = \mathrm{UGR} + \left(1.18 - \frac{0.18}{\mathrm{CC}}\right) 8 \log \left[\frac{2.55\left(1 + \frac{E_\mathrm{d}}{220}\right)}{1 + \frac{E_\mathrm{d}}{E_\mathrm{i}}}\right] \tag{2.7}$$

where CC is the ceiling coverage, taking values between 0.15–1.

The last glare index we will describe in this section is the Daylight Glare Probability index proposed by Wienold and Christoffersen (2005), defined as:

$$\mathrm{DGP} = 5.87 \times 10^{-5} E_\mathrm{v} + 9.18 \times 10^{-2} \log_{10}\left(1 + \sum_s \frac{L_s^2 \omega_s}{E_\mathrm{v}^{1.87} p_s^2}\right) + 0.16 \tag{2.8}$$

where E_v is the vertical eye illuminance [lx], L_s is the luminance of the source [cd/m^2], ω_s is the solid angle of the source [sr] and p_s is Guth's position index. They found this formula to correlate well with reported glare for DGP values between 0.2–0.8.

The DGP estimates a glare probability as a function of the vertical eye-level illuminance[8] and of the luminances of the most luminous sources in the field of view of the user, and has been derived from real, daylit windows. The inputs to this index are simple enough for this formula to be implemented in a commercial controller, provided outside solar conditions are known to the controller and some reasonable assumptions are made as to the layout of the

[8] In Wienold and Christoffersen (2005), the authors measured this illuminance in a separate room adjacent to, and identical to, the one in which the subjects underwent productivity tests. The illuminance was measured close to the position where the subject's eye would be.

user's environment (furniture, workplace disposition, etc), and as to the user's location with respect to the windows.

All these indices have been developed in an attempt to mathematically rate the glare sensation of an environment, but require a complete knowledge of luminance distributions in the field of view of the user. They are therefore difficult to use effectively in commercial controllers without at least installing a luminance mapper behind the user.

Another limitation of these formulas is that they treat only visual discomfort caused by glare, and do not address visual discomfort caused by insufficient light. Used on their own, they are insufficient for the implementation of a lighting controller but have to be complemented by normative prescriptions with respect to illuminance distributions in the field of view of the user, such as the CIE recommendations (CIE, 1986).

The formulas presented in this section attempt to quantify numerically the glare sensation or visual discomfort experienced by an average user as a function of lighting stimuli, leading to expressions that have grown increasingly complicated over the years. According to Kuhn (1996), a scientific theory that grows from an initial, simple formulation to layer upon layer of corrections and addendas is ripe for a paradigm shift, much as the geocentrism of antiquity whose simple initial expression failed to account for the observed planetary movements, and had to be completed with the unelegant theory of epicycles.

More than once, a scientific theory has been adopted over competing theories because it was the simplest, and there is an unspoken feeling among some scientists that for a theory to be correct it must be elegant. Einstein would probably add that it should also be "as simple as possible, but not simpler". Neither the UGR, GGR or any other glare formulas described in this section satisfy our definition of elegance or of simplicity, but more work in this field is certain to lead to a theory of visual comfort whose naturalness will be the best proof of its correctness.

The theories we have today require too much knowledge of the user's visual environment, and it is simply not practical to gather all this information. We do not have (yet) the necessary instruments that can evaluate the visual discomfort in realtime and feed that information to a daylighting controller. Besides, the current theories on visual comfort are statistical in nature and do not account for individual variations. The present work will, therefore, regress back to the way of empiricists and in chapter 5 we will develop a formalism for quantifying the visual discomfort by observing the behaviour of the user.

2.5 Legislative efforts

> Do you rulers ever give a just decision?
>
> *(Psalm 58:1, Good News Translation)*

In this section we will review the most important legislative documents relating to lighting comfort, control and energy management in buildings.

2.5.1 CIE Guide on Interior Lighting

The *Guide on Interior Lighting* report from the CIE (CIE, 1986) refers to the study mentioned above by Fischer (1970) where people rated workplane illuminances. The horizontal illuminance was varied from 100 lx to about 20 000 lx. Three answers were possible: too dark, satisfactory or too bright.

Illuminance range [lx]	Type of task
200–300–500	Tasks with simple visual requirements
300–500–750	Tasks with medium visual requirements
500–750–1000	Tasks with demanding visual requirements

Table 2.4: Range of illuminance prescribed by CIE (1986) for different kinds of tasks. The middle value is the optimal illuminance.

The standard concluded that:

> Practical experience has shown that an illuminance for general lighting of the order of 1000 lx is least likely to give rise to complaints, providing careful attention is paid to the avoidance of glare and to a proper balance of luminances of relevant surfaces in the room.

The standard prescribes 20 lx as the minimum required for discerning features of the human face and is therefore considered as the absolute minimum for non-working interiors. Below 200 lx, most working interiors appear unacceptably dim.

The standard ackowledges the logarithmic response of the human eye, and a 50% change in illuminance is considered the smallest increment in illuminance that results in a significant difference in subjective effect. This fact must be taken into account in tables of prescribed illuminances, as illuminance increments below this threshold are not perceptible.

The standard gives a table with recommended illuminance ranges for different tasks, given in Table 2.4.

The standard prescribes uniformity in task illuminance: the ratio of the minimum to the average illuminance of the workplane shall not be less than 0.8; the average interior illuminance shall not be less than 1/3 of the average workplane illuminance; and the average illuminance of adjacent interiors should not vary by more than a factor 5.

2.5.2 CIE Lighting of Indoor Work Places

The *Lighting of Indoor Work Places* report from the CIE (CIE-2001) is a normative effort at standardizing lighting conditions in office rooms and other workspaces. It bases its recommendations on the UGR, and gives a table that prescribes a maximum UGR, a mean workplane illuminance, and a minimum colour rendering index for each kind of activity. For office rooms, CAD workstations and conference rooms the required mean illuminance is 500 lx, the maximum UGR is 19, and the minimum colour rendering index is 80. In circulation areas the required illuminance is 300 lx, the other requirements remaining unchanged.

The daylight factor for interiors with side windows should not fall below 1% on the working plane 3 m from the window and 1 m from the walls. Supplementary lighting shall provide the required illuminance and balance the luminance distribution.

This standard furthermore adds a clause limiting the luminance of overhead luminaires that might cause discomfort glare by reflections on VDTs. For class I and II screens (high quality screens, see ISO standard 9241-7), this maximum is 1000 cd/m^2. For class III screens, this maximum is 200 cd/m^2.

The standard follows the one described in section 2.5.1 by prescribing task lighting uniformity, which shall not be less than 0.7, while the uniformity of the surroundings shall not be less than 0.5.

The logarithmic sensitivity of the human eye to illuminances is reiterated in this standard, according to which, again, a factor of 50% is the smallest significant difference in subjective effect of illuminance. This was also mentioned as early as 1937 by Luckiesh and Moss, who pointed out that the human eye is sensible to relative differences of illuminances, not absolute ones, so that a change from 50 to 100 lx is perceived as equivalent to that from 500 to 1000 lx, provided the eye had had time to adapt before the increase.

This norm has been adopted today as the european norm EN 12464-1:2002; the latter has been adopted as a swiss norm by the Schweizer Lichtgesellschaft.

2.5.3 European Parliament directive 2002/91/EC

In December 2002 the European Parliament adopted directive 2002/91/EC (EC-2002), in recognition that:

> The residential and tertiary sector, the major part of which is buildings, accounts for more than 40% of final energy consumption in the Community and is expanding.

The directive demands of Member States to establish national energy performance requirements for new and existing buildings, and to ensure that new buildings and buildings undergoing renovation are certified by independent energy experts. Boilers and air-conditioning systems must be regularly inspected. The directive does however not demand that control systems be taken into account.

2.5.4 Leadership in Energy and Environmental Design

The Leadership in Energy and Environmental Design (LEED) Green Building Rating System is a US benchmark for "the design, construction, and operation of high performance green buildings". It is a system of credits that assigns points to new or renovated buildings for meeting sustainable development goals.

An intensive use of daylight and good views to the outside are part of LEED version 2.2 for new buildings. Credits 8.1 (Daylight 75% of Spaces, 1 point) is granted if:

- At least 75% of all occupied areas have at least 2% glazing factor[9], or
- Computer simulations demonstrate that at least 75% of all regularly occupied areas have 25 fc (about 250 lx) under clear sky conditions at noon on the equinox 30 in (about 75 cm) above the floor, or
- Measurements show that at least 75% of all regularly occupied areas have achieved at least 25 fc (250 lx).

[9] The *glazing factor* is defined as $\frac{\text{Window area}}{\text{Floor area}} \times$ Window geometry factor $\times \frac{\text{Actual } T_{\text{vis}}}{\text{Minimum } T_{\text{vis}}} \times$ Window height factor, but the free LEED rating system does not define these terms—we assume they are defined in the purchasable LEED reference guide.

Credit 8.2 (Views for 90% of Spaces, 1 point) is granted if at least 90% of all regularly occupied areas have a direct line of sight to the outdoor environment via a glazing between 2'6" (76 cm) and 7'6" (229 cm) above floor finish. It is however unclear how venetian blinds whose slats are half-open should be accounted for.

2.5.5 ASHRAE 90.1-2004

ASHRAE publishes and maintains standard 90.1-2004, *Energy Standard for Buildings Except Low-Rise Residential Buildings*. The standard provides requirements for low-energy buildings and is intended to be adopted by local jurisdictions. It explicitly requires buildings of a certain size to be equipped with lighting control.

Section 9.4.1.1 of the standard (Automatic Lighting Shutoff) states that:

> Interior lighting in buildings larger than 5000 ft^2 (465 m^2) shall be controlled with an automatic control device to shut off building lighting in all spaces. This automatic control device shall function on either:
>
> - a scheduled basis using a time-of-day operated control device that turns lighting off at specific programmed times [...], or
> - an occupant sensor that shall turn lighting off within 30 minutes of an occupant leaving a space, or
> - a signal from another control or alarm system that indicates the area is unoccupied.

Section 9.4.1.2 (Space Control) states that:

> Each space enclosed by ceiling-height partitions shall have at least one control device to independently control the general lighting within the space. Each manual device shall be readily accessible and located so the occupants can see the controlled lighting.
>
> A control device shall be installed that automatically turns lighting off within 30 minutes of all occupants leaving a space [...] in classrooms [...], conference/meeting rooms, and employee lunch and break rooms.
>
> These spaces are not required to be connected to other automatic lighting shutoff controls.
>
> For all other spaces, each control device shall be activated either manually by an occupant or automatically by sensing an occupant. Each control device shall control a maximum of 2500 ft^2 (232 m^2) area for a space 10 000 ft^2 (930 m^2) or less and a maximum of 10 000 ft^2 for a space greater than 10 000 ft^2 and be capable of overriding any time-of-day scheduled shutoff control for no more than four hours.

Provided these requirements are met the designer is relatively free to choose what system will schedule automatic lighting shutoff or decide that the space is no longer occupied.

Very important
Daylight integration and control
Direct glare
Luminances of room surfaces
Reflected glare
Important
Color Appearance
Flicker
Light distribution on surfaces
Light distribution on task plane (uniformity)
Modeling of faces or objects
Shadows
Source/Task/Eye geometry
Surface Characteristics
Horizontal (500 lx) and vertical (50 lx) illuminances

Table 2.5: Elements that contribute to visual comfort, according to the *IESNA Lighting Handbook*

2.5.6 IESNA Lighting Handbook

The *IESNA Lighting Handbook* (Rea, 2000) is one of the most comprehensive handbooks available for the lighting designer.

In its chapters on visual comfort, the *Handbook* uses the VCP as its visual comfort metric since it is this metric that is predominantly used in the US. Discomfort glare is deemed not to be a problem in a building if the VCP is 70% or more. Differences of 5 units or less are not considered signficant.

In its appendices, the *Handbook* gives recommended illuminance ranges for a wide variety of activities. For "General offices, typing, computer rooms", illuminances between 300–750 lx is recommended, 500 lx being optimal.

In its section "Quality of the Visual Environment", the *Handbook* lists the elements that contribute to visual comfort in a wide range of settings, and gives their relative qualitative importance. These elements are given in Table 2.5. Note in particular the very high importance granted to daylight integration and control for general offices, except for open-plan offices where this element is considered less important.

2.5.7 Swiss norms

Swiss law requires not only that work spaces be sufficiently lit, it also requires that workers have an unobstructed view to the outside (OLT 3, art. 15, 24 and OLT 4, art. 17). Exceptions are allowed, but only if the work's technical installations prevent the workers from having their view to the outside or for special cases, such as computer rooms or photo-processing laboratories. US-style cubicles are therefore illegal in Switzerland.

Task illuminance, maximum UGR and color indices must follow the european norm EN 12464-1:2002 that we have previously discussed. The ratio between task luminance and immediate surrounding luminance must be kept lower than 3:1. The ratio between task luminance and general surrounding luminance must be kept lower than 5:1.

Zone **Hours per year**	**Usage**	**Example**	**Consumption [MJ/m^2·year]**	
			Maximum	Target
Office room 2750 h/year	300 lx, mostly daylight	Private office room	35	12
	500 lx, partly daylight	Bright office room	70	40
	300 lx, no daylight	Computer room	90	60

Table 2.6: Maximum and target values for the electrical lighting energy consumption in offices according to SIA 380/4.

Desired illuminance [lx]	**Installed power [W/m^2]**	
	Normal	Strong
50	3.2	2.5
100	4.5	3.5
200	7.0	5.5
300	10.0	7.5
400	12.5	9.0
500	15.0	11.0

Table 2.7: Recommended installed power for electric lighting. Values are given for normal and strong energy performance requirements.

The total glazed area of the building must, furthermore, be at least one eighth of the floor surface. At least half of that area must allow an outdoor view. Swiss norm SN 150911:1997 gives guidelines to building designers on how to integrate daylight in their designs.

Illuminances and glare ratings are covered in Switzerland by the european norm EN 12464-1:2002, but the SIA (Société Suisse des Ingénieurs et Architectes) publishes recommendations for the HVAC and electric lighting energy consumptions. The 380/4 recommendation by the SIA *L'Énergie Électrique dans le Bâtiment* gives maximum and target values for the electrical lighting energy consumption in different kinds of buildings. The values for office buildings are given in Table 2.6.

This recommendation also provides limits on the installed electric lighting power. These limits are reproduced in Table 2.7. Typical values for expected yearly peak consumption hours are also given, and reproduced in Table 2.8.

Zone		**Peak consumption hours [hours/year]**	
Hours per year	**Usage**	Normal	Strong
Office room 2750 h/year	300 lx, mostly daylight	1000	500
	Partly daylight:		
	300 lx	1200	900
	500 lx	1500	1100
	No daylight	2750	2400

Table 2.8: Typical consumption peak hours for electric lighting. Values are given for normal and strong energy performance requirements.

The energy requirements for heating are covered in the SIA 380/1 recommendation *L'Énergie Thermique dans le Bâtiment*. The limiting heating requirement for an administrative building are $Q_{\mathrm{h}} = 75 + 90 \times A/SRE$ [MJ/m^2·year], where A [m^2] is the ponderated sum of all surfaces of the building's envelope (ground surfaces or surfaces adjacent to unheated zones being ponderated differently), and SRE [m^2] is the *energetic reference surface*, the sum of all the building's floor surfaces (with exceptions, given in the recommendation). For a typical administrative building, $A/SRE = 0.8$ and $Q_{\mathrm{h}} = 147\,\mathrm{MJ/m^2{\cdot}year}$.

2.6 The need for an adaptive controller

Several normative documents acknowledge the difficulty of quantifying the visual discomfort for a large population. The *IESNA Lighting Handbook* (Rea, 2000), for instance, has studied the correlations between many of the glare indices described in section 2.4.3, and found that:

> All give reasonable predictions for the average discomfort of a group of people but give only poor predictions of an individual's response.

Similarly, the *Guide on Interior Lighting* report (CIE, 1986) states that:

> Because of this spread in glare sensation among people a quantity that is used to characterize the discomfort glare from a lighting installation is essentially a statistical quantity.

It seems to this author indeed very odd that the equation defining the glare indices such as the UGR involve exclusively terms external to the observer; nowhere is there any dependency on variables inherent to the personal preferences or physiology of the latter. This author had recently to take out a new car insurance and was bombarded with highly personal questions designed to tailor a very specific insurance offer that would, presumably, minimize the risk to the insurer. It is mildly amusing to note how much effort is spent on getting to know the individual by insurance companies, while no such effort is apparent in the equations describing visual comfort nor, by extension, from building or lighting designers.

The differences between individuals make it very difficult to frame a fixed set of rules that will fit all. As with thermal comfort, it is for all practical purposes impossible to satisfy everyone if the same control rules are uniformly applied. There is, however, a major difference between thermal and visual comfort. Fanger's formalism for thermal comfort (now international standard ISO 7730) is widely held to satisfy 95% of users in the ideal case, and more typically 90%—leaving only 5–10% of users dissatisfied. But US standards, as we have seen, call for Visual Comfort Probabilities to be 70% or higher, implying that our current state-of-the-art in lighting design will typically leave 30% of users dissatisfied—or three to six times as many.

Until we have a formalism for visual comfort whose satisfaction rate matches that of Fanger's, we believe that any daylighting control system should apply different rules adapted to the preferences of individual users. We believe this is the only way that satisfaction rates higher than 90% can be achieved.

2.7 Difficulties in modeling daylighting with modern blinds

A control algorithm that strives to maintain visual comfort while making an optimal use of shading devices (particularly if implemented on a low-powered, embedded controller) will work much better if it has an accurate, simple daylighting model of its lighting environment. The relative errors due to this model (i.e. the ratio between absolute error and real illuminance) must be smaller than illuminance differences the human eye can perceive.

According to Luckiesh and Moss (1937), there is a geometric relationship between illuminance and perceived effect—doubling an illuminance from 100 to 200 lx is perceived approximatively as the same change from 1000 to 2000 lx. In addition, the illuminance "must be doubled to produce an obvious and significant improvement in seeing."[10] The *IESNA Lighting Handbook* (Rea, 2000) is more conservative and gives an illuminance change of 20% as the lowest detectable change.

These elements suggest that in the absolutely worst case, a daylighting model for controllers cannot afford to be worse than 20% accurate, and 5–10% accuracy is a goal we should aim for.

Few daylight modeling methods exist today that are both accurate and computationally cheap enough to be used on embedded hardware. A ray-tracing program such as RADIANCE (Larson and Shakespeare, 2003) is excluded because of its computational cost (especially when venetian blinds are involved), and a much simpler model is needed.

The daylight factor (DF) model assumes the indoor illuminance at a given point to be proportional to the outdoor, unobstructed, horizontal illuminance for a CIE Overcast Sky (CIE-1970)[11]. This definition predicts accurate daylight illuminances only for CIE Overcast Skies. For such a sky, the indoor illuminance E is predicted by multiplying the daylight factor by the outdoor illuminance:

$$E = DF \times E_{\mathrm{g\,hor}} \tag{2.10}$$

where $E_{\mathrm{g\,hor}}$ is the outdoor horizontal illuminance and DF is the daylight factor.

The Daylight Factor method is one of two daylight illuminance calculation methods recommended by the *IESNA Lighting Handbook* (Rea, 2000). The *Handbook* recommends decomposing the DF into its sky component, its externally reflected component and its internally reflected component:

$$DF = SC + ERC + IRC \tag{2.11}$$

The *Handbook* gives tables for hand-calculating these three components for a simple rectangular room under a CIE Overcast Sky, but warns that:

> The daylight factor is a low-precision procedure for determining the illuminance at any point in an interior space produced by a sky with a known luminance

[10] This phenomenon should not come as a surprise to acousticians. The human ear is capable of operating across 12 orders of magnitude in acoustic energy. The human eye is likewise sensitive over 10 orders of magnitude. This extraordinary capacity is only possible with some kind of logarithmic response to the external stimuli—hence the geometric effect.

[11] The CIE Overcast Sky—originally known as the Moon and Spencer (1942) sky—was adopted in 1955. It is a sky whose luminance L_θ [cd/m^2] depends only on the height angle and whose zenith luminance L_{Z} [cd/m^2] is three times that at the horizon:

$$L_\theta = L_{\mathrm{Z}} \frac{1 + 2\sin\theta}{3} \tag{2.9}$$

> distribution. Direct sunlight is excluded. [...] It is used in northern Europe, where overcast skies predominate [...]

The other method recommended by the *Handbook* is the so-called lumen method, which "assumes an empty rectangular room with simple fenestration and shading devices." Furthermore:

> The procedure described here [...] does not provide for horizontal or vertical window blinds, nor for exterior elements such as sidewalks, streets, other buildings, and overhangs.

In light of these limitations, we will not describe the lumen method here.

As we will see in chapter 4, the DF model fails at predicting indoor illuminances for realistic sky conditions. Indeed, Robinson and Stone (2004) and many others argue that isotropic sky models (or models without azimuthal dependence, such as the CIE Overcast Sky) cannot accurately predict the vertical irradiance on a window. Such models do not take the sun's position into account, and those errors will inevitably propagate on the calculation of the indoor illuminance.

Guillemin (2003) has suggested that a better correlation than daylight factors might exist between indoor illuminances and vertical facade illuminance. He found experimentally that for the facades of our experimental building, the indoor illuminance could be modeled by

$$E = a \exp(b \cdot \alpha) E_{\mathrm{g\,vert}} \tag{2.12}$$

where α (between 0 and 1) is the fraction of the window not covered by its textile blind, $E_{\mathrm{g\,vert}}$ is the facade's vertical illuminance and a and b are model parameters to be fitted. For given blinds' settings, the indoor illuminance is therefore assumed to be proportional to the outdoor vertical illuminance. Guillemin reports that the errors of this model have a standard deviation of 416 lx.

We will also test this model in chapter 4 and see that it is insufficiently accurate for our needs. Moreover, this model was developed for textile blinds whereas our experiment will be carried out in offices equipped with one or two venetian blinds.

Robinson and Stone (2006) have recently proposed a simplified indoor illuminance prediction algorithm that achieves very good accuracy, in particular in the presence of reflecting neighbouring buildings. However this model does not account for the particular reflecting characteristics of venetian blinds.

Mahdavi (2001) is the only work we could find that describes a control system using a lighting simulation program, in this case, LUMINA. The paper describes a venetian blinds control system that tilts the blinds' slats according to prescribed indoor lighting conditions and is very close to fulfilling our requirements. It is however not clear whether this control system is fast enough to run on a real controlled office.

Spasojević and Mahdavi (2005) report that an excellent indoor illuminance prediction can be achieved by segmenting the sky vault in 12 sectors and measuring the luminance of each sector. The luminance of each sector is then fed into the LUMINA lighting simulation program, instead of a complete sky luminance distribution function. For an office room with two unshaded south-east facing windows, the authors report a correlation between measured and simulated indoor illuminance of $R^2 = 0.89$. But again, there does not appear to be any proof-of-concept implementation of this promising idea.

Daylighting modeling methods based on so-called Daylight Coefficients (DC) have been shown to be accurate enough to compete with ray-tracing methods, and to be computationally light. Since they fulfill so many requirements for an embedded daylight controller we will defer their discussion to chapter 4, where our own daylighting model will be described. Were it not for one important shortcoming, DC methods would presumably have been adopted in this project.

For the sake of completeness, Lehar and Glicksman (2007) have proposed an algorithm whose inputs are the geometry of the office and the distribution of reflectances in it. Each surface in the office is discretized into a mesh, and the brightness of each mesh element is given an initial brightness. The algorithm iteratively refines that guess until an equilibrium is reached and the brightness of each mesh element is a linear combination of the brightnesses of all other mesh elements. The authors found the algorithm's computation time to be about 3–5 s (i.e. too slow for our application, where tens of modellings will be made at each time step) and that "roughly the same brightness levels are reported" between their model and a RADIANCE model.

In light of this literature review it appears that no or few daylighting models exist that are fast enough to serve the needs of a real-time controller. As we will see in chapter 3, our controller will be tested against a real office room with two venetian blinds and against a virtual one with only one blind, so the daylighting model we need must be able to accomodate a variable number of venetian blinds. Such a model will be elaborated in chapter 4.

2.8 Recent control systems

In this section we will review control algorithms that have recently been proposed in the academic literature, and that can therefore be taken as the state-of-the-art in daylighting control.

Courret and Paule (1993) proposed a blinds control system that was to be fitted to the new Centre Suisse d'Electronique et de Microtechnique building in Neuchâtel. The idea was to attach artificial retinas to each facade and to classify the sky condition as either overcast, intermediate or clear.

A different control strategy is used for each sky condition. For overcast skies, the blinds are fully retracted. For intermediate skies, the blinds are completely lowered and their slats are opened. For clear skies, the blinds are lowered and the slat angle is chosen to optimize the protection against solar gains, the horizontal workplane illuminance, and visual comfort.

The slat angle is calculated to always cut off direct sun, deemed to pose too large a risk of glare. Otherwise the slat angle is chosen to provide 300 lx throughout the room's depth, and to keep the Visual Comfort Probability (c.f. section 2.4.3) above 70%.

The authors use RADIANCE to calculate the illuminance and VCP profiles for different slat angles for different sun positions. The study concludes with the computation of a daily slat angle schedule for three typical days. The study is however completely theoretical in nature. The system has unfortunately never been implemented, nor its performance assessed.

Lee et al. (1998) installed a dynamic venetian blind controller in an office room in Oakland (California), and monitored the energy consumption of that office compared to an identical one whose venetian blind was kept fixed (in the same position and slat angle) throughout the experiment, which lasted 14 months between 1996 and 1997. The daylighting controller adjusted the venetian blinds every 30 s to block direct sun and maintain the workplane illu-

minance at 540–700 lx. The system achieved lighting energy savings of 22–86% and cooling load reductions of about 28%. This study demonstrated the potential energy savings that can be achieved with a proper daylighting controller, even with one that "merely" keeps the workplane illuminance constant. Note in particular that the control algorithm makes no explicit attempt at minimizing solar gains during peak cooling periods; the cooling savings are achieved because peak cooling periods correspond usually to periods when the blinds need to be closed, resulting not only in a proper workplane illuminance but also in protection from solar gains.

The same system was then assessed with 14 real occupants (Vine et al., 1998). Surveys found that 57% of respondents found the overall lighting to be comfortable, 29% found it uncomfortable, 71% of which because of insufficient daylight.

We have alread mentioned in section 2.7 the daylight model used by Mahdavi (2001). In his paper, he also simulates the operation of a controller that uses this daylight model. His controller calculates at each timestep the value of a *utility* function for each combination of slat angle and electric lighting power, the optimal combination being then selected. This use of a utility (or cost) function is also the basis of the daylight controller we have developed in this work. It was also used by Ferguson (1990) in her thermal control algorithm.

Guillemin (2003) has proposed a daylight control algorithm for office rooms equipped with two textile blinds based on a set of fuzzy rules. The variables that define the external environment (outdoor temperature, vertical facade illuminance, etc.) are discretized into categories and each possible combination is assigned an optimal blinds' position. He found that 18 rules were necessary for a typical office room. The values of each rule are optimized daily so the controller can reproduce as closely as possible the blinds' positions chosen by the users themselves.

This algorithm is distinct from the previous ones in that it does not attempt to keep the controlled environment close to a design value, or the illuminances to prescribed values, or a glare index at a minimum. Instead, an expert judgment assigns initial values to each rule. If and when the user overrides the system, these rules are accordingly adjusted. This algorithm was shown to yield about 25% energy savings and to be accepted by 95% of the surveyed users (about 20 in all). The algorithm was implemented on the LESO experimental building (c.f. section 3.1), the same building on which the controller described in this work will be implemented.

Variations on the theme of fuzzy logic control have been proposed by Lah et al. (2006) and Kolokotsa et al. (2006). In both cases, the control of a building is done through a set of fuzzy rules given initial values by a human expert. The work by Kolokotsa et al., in particular, is very close to the work described by Guillemin in that the fuzzy logic controller is implemented in MATLAB and controls a building equipped with a commercial EIB system, which is exactly the conditions in which Guillemin worked. However, neither of these systems carries out any optimization of the rules nor takes into account user overrides.

In spite of such promising results by machine-learning control algorithms, no commercial control system exists today that implements any of these ideas. Some innovative products are being rumoured, but even the daylight control system that will equip the New York Times Headquarters building, currently under constrution (Lee and Selkowitz, 2006), will limit itself to preventing direct sunlight and complement daylight with artificial lighting until a suitable workplane illuminance is reached (484–538 lx).

The control algorithm described in this work is the first venetian blinds and electric lighting control algorithm proposed by our institute. Its underlying philosophy is different from that of

the control algorithms described in this section, although it too attempts to minimize energy consumption while adapting itself to the user's preferences.

2.9 Scope of this project

This work was carried out as part of the Ecco-build project, whose goal was to develop and implement an adaptive control algorithm for venetian blinds and electric lighting, that minimizes the office room's lighting and heating electrical consumption while maintaining the user visual comfort. We have implemented this algorithm on a pre-existing, real, occupied office while making sure it can also be run against a computer simulation of an office. The control algorithm will keep the office room's energy consumption to a minimum, but learn from user overrides.

No particular assumption has been made about the geographical location or orientation of the office, nor about its geometry or the distribution of its furniture. The location is an input to the algorithm but only for calculating the sun's position. No parameterized model has been used in the visual discomfort estimation nor in the controller's daylighting model, which is deduced from first principles.

The controller has been implemented in a popular programming language used in embedded controllers. A limited number of mathematically complicated modules will use a compiled external mathematical library. Throughout this work we will not only discuss its theoretical principles but always keep an eye on a possible implementation on a commercial embedded controller.

3 Monitoring and simulation of controlled office rooms

> "What more do you want to know?"
> "The names of all the stars, and of all living things, and the whole history of Middle-Earth and Over-heaven and of the Sundering Seas," laughed Pippin. "Of course! What else?"
>
> *(J. R. R. Tolkien, The Two Towers)*

In this chapter we will describe not only the office rooms on which our controller has been implemented, but also the computer model that helped develop and implement it, as well as assess its energy performance in different situations.

Section 3.1 begins by describing the LESO solar experimental building, on which our controller's implementation runs today. We will describe the two building office rooms that were fitted for that purpose and describe the hardware that has been installed.

Section 3.2 then describes the computer model of a typical office room that was developed within this project, and that has been used for the development and assessment of the control algorithm and its implementation. As we will see, the computer program that implements the algorithm described in this work was designed to run indifferently against a real or a virtual office room. This is possible because that program is completely decoupled from the interface to the building automation system, which is replicated by the simulation program.

3.1 Office rooms description

The LESO solar experimental building (Figure 3.1) is a middle-size (1000 m^2) administrative building with 20 south-oriented office rooms of 15.7 m^2 floor area each and height 2.8 m. Approximately half of them have a single user and the other half two. The south facade of the building was refurbished in 1999 with one that would satisfy sustainable development requirements, being made principally of wood and integrating advanced daylighting systems (so-called anidolic systems) that redistribute daylight in the offices (c.f. Figure 3.2).

The LESO building is divided in nine thermically isolated units (three per floor). Most units have been partitioned in two office rooms. The conference room (LE 2 205) and the workshop (LE 0 05) are exceptions and occupy a complete unit each. The materials making up LESO's construction elements are given in Table 3.1.

The doors are 3.0 m^2 in area and made of 2 cm thick wood. The lower windows are 2.1 m^2 in area. The upper (anidolic) windows are inclined 45° and are 1.7 m^2 in area. All windows (Silverstar N 13020 S glazing) are double glazed with IR coating, U-value 1.4 $W/m^2 \cdot K$ and g-value 0.54. The windows' frame is made of wood and its area is 0.9 m^2, its U-value 2 $W/m^2 \cdot K$.

The monitored total energy intensity of the building (including electric heating, lighting, computers and machines) is equal to 232 $MJ/m^2 \cdot$year, of which 76 $MJ/m^2 \cdot$year are needed for

Figure 3.1: The LESO solar experimental building in Lausanne, Switzerland.

Figure 3.2: The anidolic system of mirrors that redistribute daylight in LESO's workshop.

Element	**Type**	**Area [m^2]**	**Material**	**Thickness [cm]**
South facade wall	Light wall	5.4	Plaster panel	1
			Thermal insulation	12
			Wood	1
North wall (towards circulation area)	Heavy partition wall	7.0	Concrete bricks	12
			Thermal insulation	8
			Concrete bricks	12
Wall between units	Heavy partition wall	13.3	Concrete bricks	12
			Thermal insulation	8
			Concrete bricks	12
Wall within unit	Light partition wall	13.3	Plaster panel	1
			Thermal insulation	4
			Plaster panel	1
Floor		15.7	Rubber coating	1
			Screed	6
			Thermal insulation	6
			Concrete slab	25
Ceiling (last floor)		15.7	Concrete slab	25
			Thermal insulation	16
			Concrete and gravel	10

Table 3.1: Materials and thicknesses of LESO construction elements. The external (west and east) walls have a 16 cm concrete bricks layer instead of 12 cm.

heating space and 42 MJ/m^2·year for lighting. The building is equipped with a grid-connected photovoltaic installation on its flat roof which provides 15.4 MJ/m^2·year. Altherr and Gay (2002) have reported that the energy signature of the building when the outside temperature drops below 13°C (the so-called temperature of heating interlocking) is 0.72 W/m^2·K. Above this temperature, the free (casual and solar) gains of the building are sufficient to heat it. The building has no active cooling or ventilation system except in one computer room. It is otherwise naturally ventilated by a stack effect.

A detailed description of the building with its new southern facade, including an exhaustive analysis of the building's energy flows, is given by Altherr and Gay (2002).

The tasks carried out in these office rooms is evenly split between reading, writing, and work on the computer. The VDTs are mostly flat-screen monitors. The windows of all offices (except the ones used in this experiment) are protected by two external textile blinds. Floor drawings of the three building floors are shown in Figure 3.3.

This building was used before its refurbishment for the experimentation of passive solar systems. Today it serves as test bench for advanced control algorithms for building services (heating, blinds and electric lighting). It is equipped with a commercial off-the-shelf European Installation Bus (EIB) building management system, with the following sensors and actuators:

Outdoor sensors: outside air temperature, horizontal diffuse and global radiation, horizontal and south vertical illuminance, wind speed and direction.

Indoor sensors (each office): air temperature, horizontal workplane illuminance, occupancy, and window opening.

Indoor actuators (each office): blinds, electric lighting (continuous dimming), and electrical heating.

User interface (each office): blinds, electric lighting, and indoor temperature setpoint.

The electrical consumption of each room is also recorded. Each electrical meter has a rotating metallic disk on which a small (about 10°) sector is painted black. This disk is lit and rotates in front of a photodiode. When the black sector passes in front of the photodiode, a contact is broken, which is detected by a binary contact sensor connected to the EIB.

The heating ("Force" meters) is recorded separately from the other appliances ("Lumière" meters). Most Force meters are rated at 120 revolutions per kWh, and most Lumière meters at 480 revolutions per kWh. No details are given on the meters about their precision, and we assume therefore conservatively an uncertainty of ±10 revolutions per kWh for both kinds of meters, i.e. a relative uncertainty on the energy consumption measurement of 8% for the Force meters and 2% on the Lumière meters.

The EIB building management system was installed in 1999. A PC is connected to that bus through a EIB-serial interface in order to run the control and monitoring program described in section 3.1.1. Since then, two research projects focused on control systems have been carried out using mesurements on the building, AdControl (Guillemin, 2003; Guillemin and Morel, 2003; Guillemin and Molteni, 2002) and Ecco-build. A data acquisition program has been running independently of any research project continuously since 2002 (except for upgrades). The data is stored in a MySQL database (MySQL Development Team, 2004, c.f. section 3.1.3), representing slightly more than five years of monitoring. For data related to user actions alone (change of occupancy, manual use of blinds or electric lighting controls, etc), we have accumulated at the time of writing about three million datapoints.

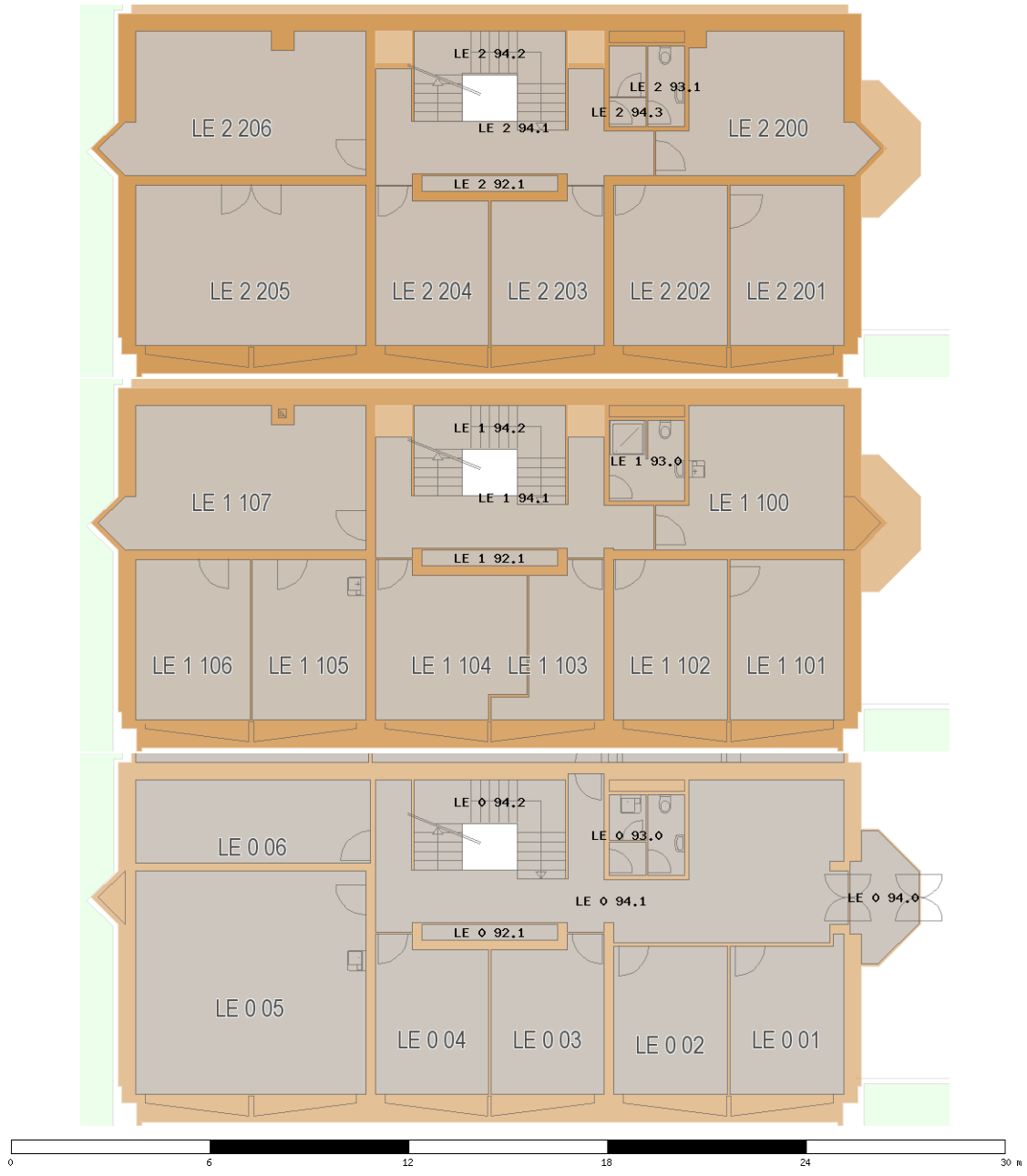


Figure 3.3: Floor drawings of the LESO building. (0: basement; 1: first floor; 2: second floor.)

Figure 3.4: LESO office rooms equipped for this project. Office room 201 is to the left, office room 202 to the right.

Two adjacent offices rooms (c.f. Figure 3.4) located on the second floor of the LESO building were equipped with venetian blinds in July 2004 and used in this study. The lower window was fitted with external blinds Baumann-Hüppe type "Noval 90", bicolor slats RAL9016/RAL7035. The upper window was fitted with indoor[1] blinds Baumann-Hüppe type "Genius 50" (c.f. Figure 3.5).

Table 3.2 lists the devices installed in each of these two office rooms, as well as the equipment that monitors the outdoor environmental conditions.

The horizontal workplane illuminance is measured with Siemens brightness sensors GE 252, which are actually ceiling-mounted luminance sensors shielded from the window's luminance. The conversion from the workspace's luminance to its illuminance is a programmable feature of the sensor, so that it can be calibrated to determine the illuminance, assuming a constant reflectance in its visual field. Each sensor has been calibrated with a reference sensor by Guillemin (2003). The sensors' output is linear only up to about 500 lx, and has to be software-corrected in order to be accurate up to 3500 lx.

Three additional illuminance sensors were installed in each office room used in this project, Wienold and Christoffersen (2006) having reported a good correlation between the glare sensation due to daylight and the eye-level vertical illuminance. Our control algorithm, described in chapter 5, is able to combine information from different sources and will adapt faster to the user's preferences with three illuminance sensors instead of one. The first sensor was vertically installed at eye-level on the rear wall, slightly next to the user's usual sitting place. The second one was placed on the opposite wall. The third sensor was placed horizontally about 15 cm above the user's desks, giving an accurate reading of the workplane illuminance.

The luxmeters provide an output current proportional to the intensity of the light flux impinging on them. A 100 Ω resistor was used to load them, allowing to read out a voltage (instead of a current). We calibrated them by placing their sensor under an adjustable light source, next to a reference illuminance sensor. The light source was gradually increased up to about 50 000 lx and the corresponding voltage was read out. The calibration curves are given in Figure 3.11. The sensors were then installed and connected to the GIRA Giersiepen analog sensor interface. One such sensor is shown in Figure 3.9.

[1] The existing outdoor textile blinds, and the upper window's slope, prevented us from fitting outdoor venetian blinds to the upper windows (which would have been a superior setup, from a thermal point of view).

Figure 3.5: The blinds equipping office room 202. The upper indoor blinds are Baumann-Hüppe type “Genius 50”, the lower outdoor blinds are Baumann-Hüppe type “Noval 90”.

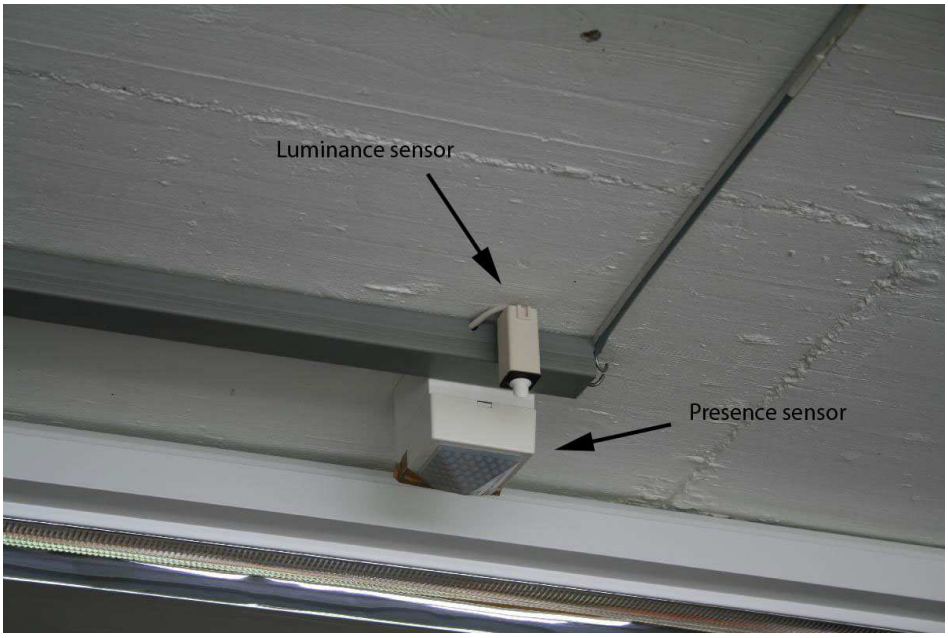


Figure 3.6: Presence and ceiling-mounted luminance sensor.

Manufacturer	**Model**	**Type**	**Units**	**Error**	**Figure**
In each office					
High Technology Systems	AG ECO-IR 360AC-EIB	Presence sensor	Boolean		3.6
Siemens	UP 231/2 Delta profil	Push-button interface and temperature sensor	°C	±0.5°C	3.8
Siemens	GE 252	(Il)luminance sensor	lx	15%	3.6
BBC Goertz/Metrawatt	MX4	Illuminance sensor	lx	5%	3.9
GIRA	Analog-Sensorschnittstelle 4fach	Analog/digital converter, 4 channels			
Siemens	N 562	Binary output (for heating control)			
Siemens	UP 220	Push button interface for blinds control			3.7
ABB	JA/S4.230.1	Blinds actuator			
		Force meter	kWh	8%	
		Lumière meter	kWh	2%	
Outdoor devices					
Delta-T	BF3	Sunshine sensor	W/m^2	12%	3.10
GIRA	Analog-Sensorschnittstelle 4fach	Analog/digital converter, 4 channels			
Phoenix Contact	MCR/PT100	Temperature sensor	°C	±0.5°C	

Table 3.2: Devices installed in office rooms 201 and 202 at LESO. Outdoor devices are also given.

Figure 3.7: Venetian blinds manual controls.

Figure 3.8: Electric lighting and temperature setpoint control box. The user sets the electric lighting's dimming levels with the two middle rockers. The leftmost rocker is used to select the temperature setpoint. The current temperature setpoint offset from 20° is indicated by the LED on the horizontal scale. The rightmost rocker is unused.

Figure 3.9: Wall-mounted illuminance sensor EB3 in office 202.

Figure 3.10: Delta-T BF3 sunshine sensor. A shading mask ensures that at least one of seven photosensitive diodes is always lit by direct sunlight, and at least one is shaded from it. The sensor can thus determine the diffuse and global components of daylight, and does not require any polar alignment (Delta-T Devices Ltd, 2006).

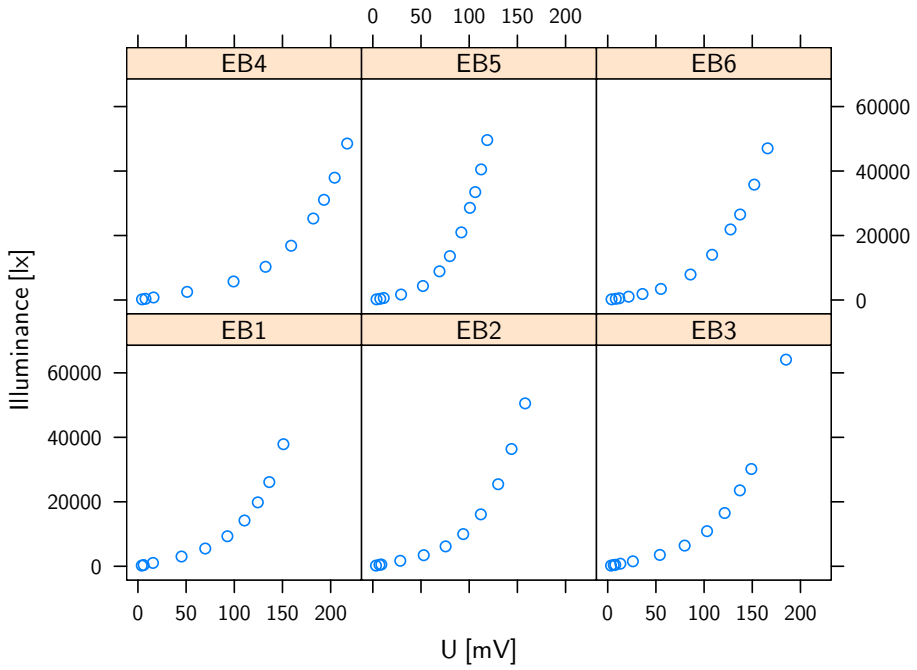


Figure 3.11: Luxmeters calibration curves for the six luxmeters installed in office rooms 201 and 202. The luxmeters are labeled EB1, EB2 and so forth.

3.1.1 The Eibserver program

The LESO building is equipped with a commercial building management system that can read out the status of the building's sensors and actuators, and send commands to the latter. It was installed in 1999. As of August 2004, 240 sensors and actuators were on this system.

Following the intial work of René Altherr, we have extended and currently maintain EIBSERVER, a Java program that allows a PC connected to the EIB system via its serial port to listen to, and send, EIB telegrams. It keeps in memory at all times the complete known state of the LESO building—when any variable is modified, that change is committed to memory and logged to disk (see section 3.1.2).

EIBSERVER implements the Java Remote Method Invocation (RMI) mechanism to allow other processes, possibly running on remote machines, to subscribe for notification on specific events, to query the building's status and to send control commands of their own. This can be done by any program implementing a Java virtual machine, including Matlab. But Matlab is unlikely to run on embedded devices in the foreseeable future (both for technical and economical reasons), whereas many Java virtual machines for embedded environments exist today[2]. This is why we have tried to write as much of our controller in pure Java, as discussed in chapter 6.

Below we give an example script written in Jython, an implementation of the Python programming language written in Java, with which we interactively query and send commands to the EIB system. Commands entered by the user are in bold:

[2]E.g., `http://www.skelmir.com/` or `http://k-embedded-java.com/`.

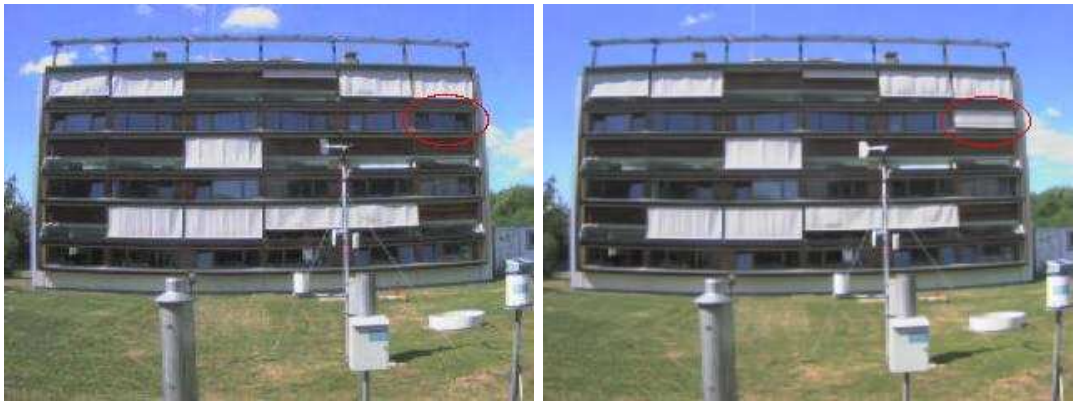


Figure 3.12: Interactive manipulation of textile blinds. The left photo is before sending the command, the right is after. The blind that moved is circled in red.

```
1  [lindelof@lesopriv3 ~]$ jython
2  Jython 2.2a0 on java1.4.2_05 (JIT: null)
3  Type "copyright", "credits" or "license" for more information.
4  >>> from java.rmi import Naming
5  >>> blind_201 = Naming.lookup("//lesopc7/bld-low_201")
6  >>> blind_201.getPosition()
7  1.0
8  >>> blind_201.setPosition(0.2)
9  >>> blind_201.getPosition()
10 0.2031
11 [lindelof@lesopriv3 ~]$
```

On line 5 we query the RMI server running on the same machine as EIBSERVER (i.e. `lesopc7`) for the remote object registered under the name `bld-low_201`, which is the lower textile blind in office 201. In the next lines the blinds' position is queried, then set to 20% window's fraction open, and queried again. The results, as seen by a webcam pointed at LESO's south facade, are shown in Figure 3.12.

As we will see, the remote interface exposed and implemented by EIBSERVER will also be implemented by the building simulation program described in section 3.2. Because this interface is common, it will be fully described in its own section 6.3.

3.1.2 Data logging

The EIBSERVER program logs all bus events to disk. It first creates, if they do not exist already, subdirectories called `lighting/`, `occupancy/`, and so on in a `results/` directory.

In each of these directories, a new text file named `DDMMYY.res` (possibly appended with one or more tildes (`~`) if EIBSERVER had to restart that day) is created every night at midnight. The EIBSERVER program appends to these files the data as it comes in.

The format of the data is, however, not quite consistent and has obviously evolved over time. Presently, according to comments left by the EIBSERVER program's original author, that format is `DDMMYY:HH:MM:SS;device_room;value` where `device` is the name of the devices,

Table name	Description	Records
`luminosity`	Indoor illuminance [lx]	26 946 119
`skyluminosity`	Outdoor illuminance [lx] / irradiance [W/m^2]	20 994 794
`consumption`	Electrical meters [kWh]	5 238 006
`occupancy`	User presence [boolean]	2 986 328
`temperature`	Indoor and outdoor temperature [°C]	2 722 403
`blind`	Textile and venetian blinds' positions and slat angles [0–1]	976 656
`heating`	Electrical heating power [%]	803 075
`lighting`	Artificial lighting power [%]	379 339
`window`	Windows open/closed status [boolean]	64 104
`setpoint`	User-chosen temperature setpoint [°C]	5 508

Table 3.3: The tables in the '`leso_eib`' database. The rows are sorted by decreasing number of records. The table names are identical to the names of the data directories created by EIBSERVER. The number of records are as of April 2007.

possibly prefixed with `usr:` or `set:`, and possibly suffixed with `_stat`, `_pos` or some other descriptive string. `room` is the room number (3 digits), and `value` is the numeric data.

Note in particular that according to the prefix of the device's name one knows whether it was a user action or a command sent through EIBSERVER. It is thus possible to distinguish controller commands from user commands.

Early data, however, presents difficulties. In rare cases the line is unrecoverable gibberish. Occasionally, race conditions allowed a thread to write out its data while another one was looking up the operating system's definition of a newline, yielding two data entries on the same line followed by two newlines. These problems were rare and have not occured anymore.

3.1.3 The '`leso_eib`' database

We have written a Perl script (see Appendix C.1 for the source code) that starts every night at 5 a.m.[3], reads in the logfiles that have been written that day since midnight (one per subdirectory), parses each line, and inserts the corresponding values in a MySQL database. After reaching the end of each file the program checks every minute whether new data is available in that file, and updates the database accordingly. It stops at midnight. The MySQL database includes also all the data recorded by EIBSERVER since it began recording data in 2002.

The database is made up of ten tables (sharing an identical structure), corresponding to the ten categories of data that EIBSERVER records. These tables and their schema are given in Tables 3.3 and 3.4.

The database is accessible from a web interface, available on the EPFL intranet from `http://lesopc28.epfl.ch/~lindelof/phpMyAdmin/index.php`. From this interface users have read-only access to the entire database, and can dump portions of the data as text files. The main interface is shown in Figures 3.13 and 3.14. The database can, of course, also be programatically accessed.

[3] All textile blinds are fully retracted by EIBSERVER at 2 a.m. and lowered again to their previous position. This ensures that any drift in the measurement of the blinds' position is kept to a minimum. The Perl script starts at 5 a.m. to be sure that no other "administrative" process runs on the LESO building.

Field	MySQL Type	Description
index	int(11)	Record index
date	date	Date of the record (YYYY-MM-DD)
time	time	Time of the record (HH:MM:SS)
room	char(3)	Office number (3-digit number, 300 for outdoor events)
type	varchar(5)	Source of the event (user, controller, sensor)
device	varchar(20)	Device name
action	varchar(5)	Kind of event (sensor, actuator command, etc.)
data	double	Numeric data

Table 3.4: Schema of the '`leso_eib`' database's tables.

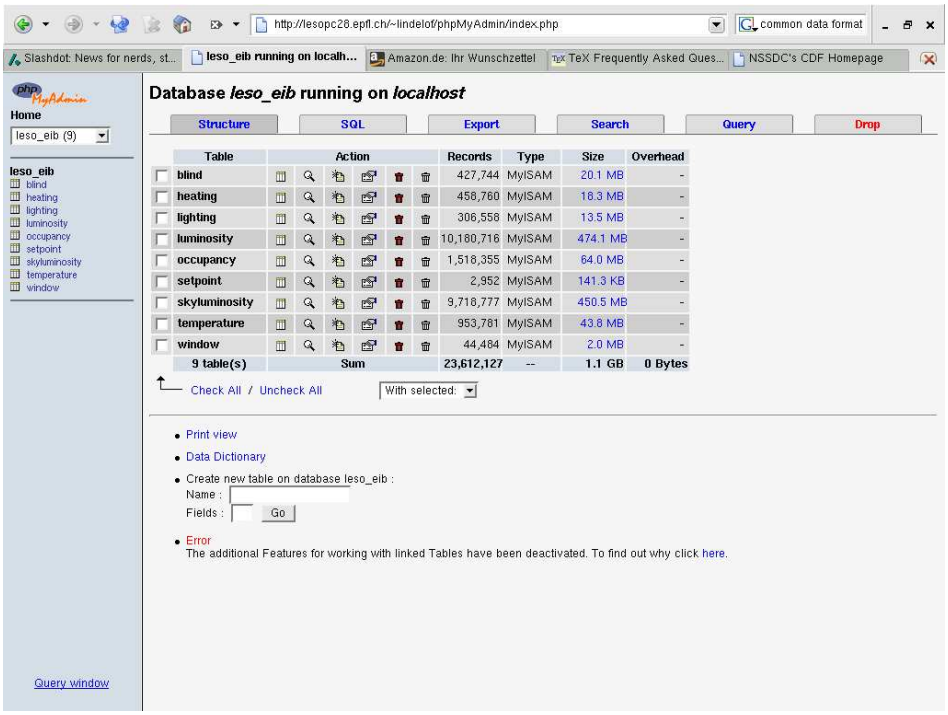


Figure 3.13: The main interface to the '`leso_eib`' database.

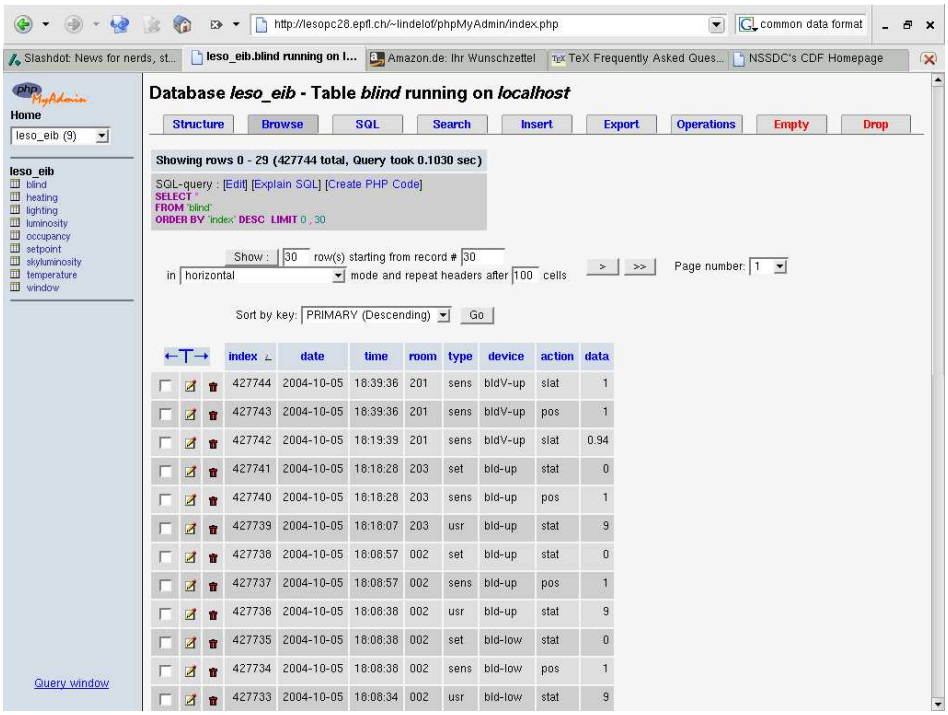


Figure 3.14: Browsing a table of the '`leso_eib`' database.

3.2 Simbad model

One of the goals of this project is to demonstrate through computer simulations the potential energy savings with a blinds and electric lighting controller. To that end it was important to choose and, if need be, extend a building simulation package.

Many software packages exist that model and simulate the thermal behaviour of building elements. The U.S. Department of Energy's website[4] lists at the time of writing 344 of them. ESP-r[5] and EnergyPlus[6] are the most popular packages in Europe and USA respectively.

In this project we needed a software package that would model the thermal and visual aspects of an office room whose electric lighting and blinds are computer-controlled. We also required that the existing controller implementation could be easily coupled to this simulator.

SIMBAD (SIMBAD-2006) is a MATLAB toolbox for building simulation developed by the french Centre Scientifique et Technique du Bâtiment (CSTB). It models most HVAC systems within a building but comes with no accurate daylighting model for a complex system such as venetian blinds. SIMBAD had, however, two very desirable features.

First, as a MATLAB toolbox, it is a library of simulation blocks for Simulink, the graphical tool for modeling and simulating dynamic systems under MATLAB. Assembling a building model is very easy in this environment. SIMBAD provides out-of-the-box blocks for many building appliances such as heating elements or ventilation systems.

Second, Simulink runs under MATLAB, which has its own Java virtual machine. Java classes can therefore be loaded in MATLAB and interact with the MATLAB environment. Since

[4] `http://www.eere.energy.gov/buildings/tools_directory/`
[5] `http://www.esru.strath.ac.uk/Programs/ESP-r.htm`
[6] `http://www.energyplus.gov/`

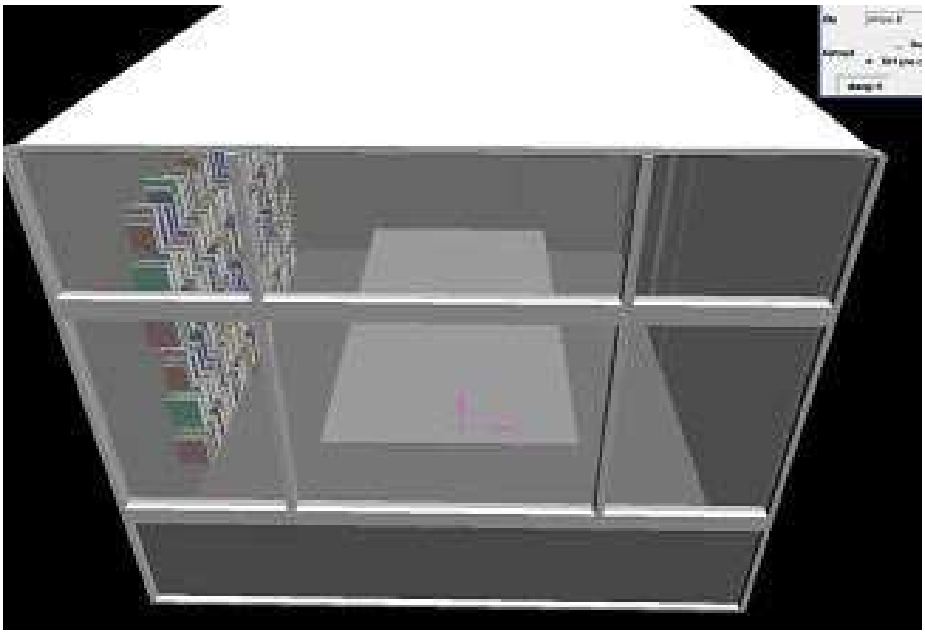


Figure 3.15: RADIANCE model of the virtual office room.

our controller's implementation is completely decoupled from the interface to the building management system, we can write an interface in Java to our SIMBAD model that masquerades as the real building, and our controller will not see any difference.

The benefits of this approach cannot be understated. Again: as long as we have a well-defined interface to the controlled systems, we are free to implement any control algorithm in any dialect of Java. In particular, we can write prototypes in Python using Jython[7] (a Python implementation written in Java), or Groovy[8] (a scripting platform for the Java language). This implementation can then run against EIBSERVER or against SIMBAD, since they both present this well-defined interface. As we will see, this led to dramatically short development cycles where programming defects were quickly discovered with the simulator and fixed before the real office room's controller software was updated. The interface used in this project is described in section 6.3, but this idea of a common controller interface for any building management system will be elaborated further in Appendix A.

The full details of our office room model are given by Marty and Fontoynont (2006). In the following we will discuss the elements of this model most important to us: the daylighting model, the user behaviour model and the heating/cooling model. The interface to the controller will be described in section 6.3.

3.2.1 Daylighting and electric lighting model

The virtual office room is a cuboid $4.61 \times 3.62 \times 2.85\,\mathrm{m}^3$ in volume, with one side almost fully glazed and protected with venetian blinds. Figure 3.15 shows the RADIANCE model of this office room.

[7] `http://www.jython.org/Project/index.html`
[8] `http://groovy.codehaus.org/`

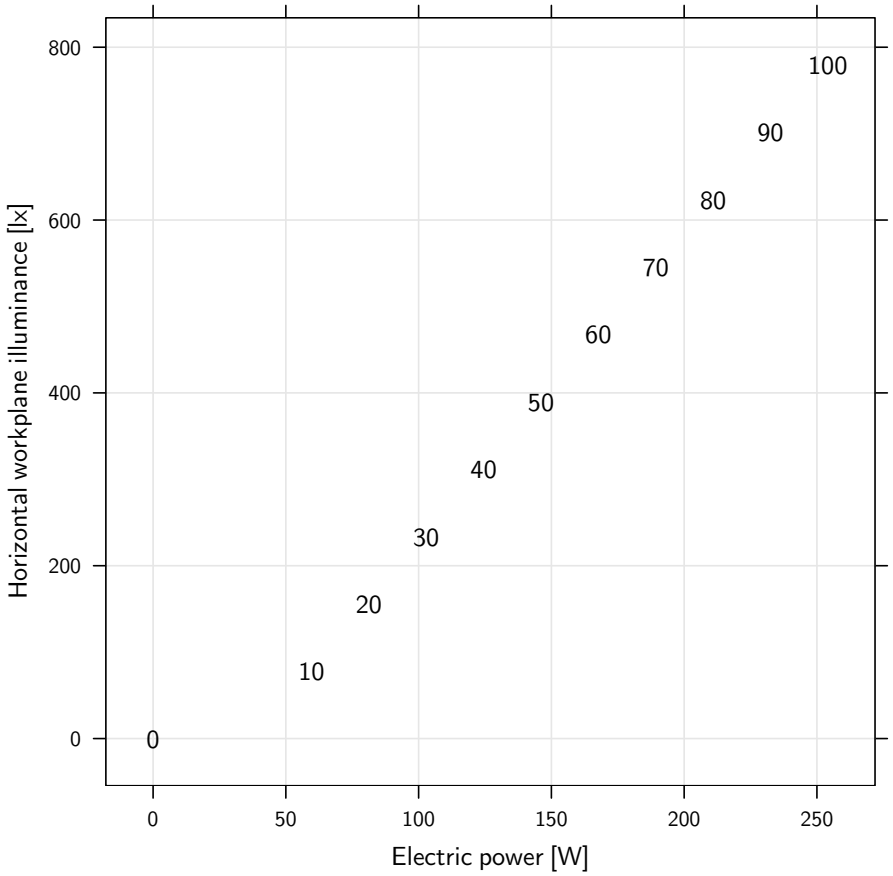


Figure 3.16: SIMBAD electric lighting calibration. The numbers represent the lighting command in % sent to the SIMBAD block.

The electric lighting model consists of six 36 W fluorescent tubes (or 12.94 W/m^2 installed power density) with a 95 lm/W efficiency and a 20% time depreciation factor. The ballasts in this model are assumed to be responsible for a 15% overconsumption (about 30 W). The office room is 16.7 m^2 in area and has an 80% utilance factor, so the electric lighting can provide an illuminance anywhere between 0–780 lx[9]. This high upper limit was deliberately chosen to see whether the controller would correctly limit the illuminance to more reasonable levels.

We have used a Python script to increase the lighting command in steps of 10%, and read out the electric power consumption and resulting illuminance. The result is shown in Figure 3.16.

For the daylighting model, we briefly considered a daylight coefficients approach[10], whereby we would compute the coefficients for every blinds' position, slat angle and illuminance sensor position combination. We would then also have had to implement a sky model in SIMBAD, but because of time pressures we decided instead to precalculate every possible illuminance

[9] $6 \times 36 \times (1 - 0.2)/16.7 \times 80\% = 786$ lx.

[10] Daylight coefficients will be discussed again together with the controller's daylight model, and defined in chapter 4.

Column	Units	Description	Example
1	NA	Month (January=1)	6
2	NA	Day of the month	4
3	NA	Hour of the day (with fractions)	9.292 (09:17:30)
4	lx	East wall illuminance	255
5	lx	West wall illuminance	714
6	lx	East eye-level illuminance	2584
7	lx	West eye-level illuminance	3064
8	lx	Workplace illuminance	1373

Table 3.5: Precalculated illuminance file format. The examples come from file `illum_v001_0_0_N.ill`. The wall illuminances are calculated 2.1 m away from the facade, at a height of 2.15 m. The eye-level illuminances are calculated 2.1 m away from the facade, at a height of 1.2 m. The workplane illuminance is calculated 2.1 m away from the facade at a height of 0.8 m.

for each timestep, office room location, office room orientation and blinds' settings.

This RADIANCE-based daylighting model consists of precalculated illuminances at five different positions in the virtual office room, for each minute of a year, with weather data files for Brussels (Belgium, 50°52' N 4°22' E) and for Rome (Italy, 41°54' N 12°27' E). These annual simulations were carried out for three different office orientations (south, west, and north). A roller blind or a venetian blind was added to the simulated model, and different simulations were made for different discrete blinds' settings.

785 annual simulations were carried out, each yielding the five illuminances for every minute for the 525 600 minutes in a year. Each file is 32 megabyte large, and the whole data set represents 24 gigabytes. Jan Wienold of Fraunhofer-ISE carried out the simulations with the RADIANCE-based DAYSIM (Reinhart, 2006) program on a cluster of more than 50 nodes. The weather data was provided by Fraunhofer-ISE, who produced it with the Metenorm (METEONORM-2006) program. DAYSIM models the sky luminance distribution from global and direct outdoor horizontal illuminances with Perez's all-weather sky model (Perez et al., 1993).

Table 3.5 gives the structure of these precalculated illuminance files. A full illuminance simulation with meteorological data thus yields a table with 535 600 rows, whose structure is given in Table 3.6. The naming scheme for these files in given in Appendix C.2.

3.2.2 User behaviour model

The virtual user is present in the office from 8 a.m. to noon, and from 2 p.m. to 6 p.m. during weekdays, and absent on weekends. No other holidays are modeled.

The SIMBAD model runs with either the simulated user controlling the blinds and the artificial lighting, or the external controller, but not both. The user does not act on the office room controls at all while the automatic controller runs, but follows the same occupancy schedule.

When the user controls the blinds, an algorithm, that follows a recent state-of-the-art review of users' manual blinds control (Marty et al., 2003; Reinhart and Voss, 2003), determines the blinds' position and slat angle:

Field	Units	Description	Example
`index`	N/A	Datapoint index	300000
`time`	POSIX time	Timestep	2006-07-28 08:59:30
`Eeg`	W/m^2	Horizontal global irradiance	341
`Eed`	W/m^2	Horizontal diffuse irradiance	118
`Eev`	W/m^2	Vertical global irradiance	67.6
`Ees`	W/m^2	Direct normal irradiance	515
`temperature`	°C	Outside temperature	16.0
`rightWall`	lx	Right wall illuminance	249
`leftWall`	lx	Left wall illuminance	629
`rightEye`	lx	Right user position eye-level illuminance	2542
`leftEye`	lx	Left user position eye-level illuminance	3007
`horiz`	lx	Horizontal workplane illuminance	1398
`elevation`	°	Solar elevation	25.7
`azimuth`	°	Solar azimuth (0° is north, 90° is east)	91
`clearness`	N/A	Perez clearness ϵ	2.76
`clearness.cat`	N/A	Perez clearness category $\tilde{\epsilon}$	5

Table 3.6: Structure of the simulated data. The Perez clearness ϵ and clearness category $\tilde{\epsilon}$ will be defined in chapter 4. The names of the fields are the same as the ones used in the analysis code (Appendix C.1).

- At dawn and sunset, the blinds are completely up.
- If direct sun is present, the slat angle is adjusted to the cut-off angle (adding 5° if the slat angle was less than the cut-off angle). The height is adjusted so the sun patch is at least 50 cm away from the user.
- If the vertical illuminance on the facade is larger than 49 klx, the blinds are pulled completely down and the slat angle is adjusted to the cut-off angle (adding 5° if the slat angle was less than the cut-off angle).

When the user does any of these, no more actions can occur for 15 min. For these purposes, direct sun is considered present if the ratio global/diffuse irradiance is larger than 1.25 and the horizontal global irradiance is larger than $24\,W/m^2$. This definition is used internally by our BF3 sunshine sensor, and is known (Delta-T Devices Ltd, 2006) to correlate well with the World Meteorological Organisation's definition (beam irradiance $> 120\,W/m^2$).

The electric lighting is switched on by the user to maintain a desktop illuminance of at least 500 lx. The user can adjust the electric lighting power in steps of 33%. The user never decreases the electric lighting power before leaving the office (at noon or in the evening), when the lighting is switched off.

This might sound harsh, but we have recently studied the so-called *intermediate light switching probability* by the LESO users, i.e. the probability as a function of indoor illuminance that the users switch their lights off without leaving the office. That study (Lindelöf and Morel, 2006) found that whereas the intermediate switch-on probability for a 15-min timestep can be between 1% and 3% for workplane illuminances lower than 200 lx, the switch-off probability

Name	Description	Units
Inputs		
T°ext	Outdoor temperature	°C
CMV flow	A vector representing controlled mechanical ventilation and infiltrations	
H/C flux	Heating/cooling loads	W
Int gains	Internal heat gains	W
Solar flux	Solar heat gains	W
Outputs		
Air	A vector containing the characteristics of the room's air, including its temperature	
T°w	Wall temperature	°C

Table 3.7: Inputs and outputs to SIMBAD's thermal model.

does not depend on the illuminance and is lower than 0.2%. That study found also that most LESO users did not bother dimming their lights but turned them either fully on of fully off.

A modeled user that never switches the lights off before leaving the office is, therefore, not worse than the average LESO user[11]. Allowing the modeled user to dim the lights is, energetically speaking, an improvement. We therefore believe that our user model is fair, or even optimistic, and that our controller will be compared with an energy-conscious manual behaviour, not with a hopelessly irrational and wasteful one.

3.2.3 Heating and cooling model

An 1000 W electric heater was added to the office model with a simple, independent controller that turns it on if the temperature decreases below 21° and turns it off when the temperature increases above 23°[12]. The heater's electric consumption is separetely recorded.

A fancoil with water alimentation was also added to the model for cooling. It switches on if the temperature goes above 26°, and increases in power according to the current temperature. Its power consumption is also recorded.

3.2.4 Thermal model

The office room's thermal behaviour is modeled with a SIMBAD two-node thermal 'Office room, medium inertia and high insulation' block. One node is the indoor air temperature, the other is the wall temperature. The inputs and outputs of this model are given in Table 3.7.

For each node, SIMBAD solves at each timestep the following equation:

$$C_i \frac{\mathrm{d}T_i(t)}{\mathrm{d}t} = -G_i \Delta T(t) + q_i(t) \tag{3.1}$$

where $T_i(t)$ [K] is the temperature of node i, C_i [J/K] is the heat capacity of node i, G_i [W/K] is the heat flow conductance between the two nodes, and ΔT [K] is the temperature difference

[11] The average LESO user will, however, leave the office room more often than only twice a day, increasing the probability of switching the lights off when they are not needed.

[12] Note that more energy-conscious values would be 19° and 21° respectively.

between the nodes. $q(t)$ [W] is the sum of the solar or casual gains, and of the heat losses, at node i, including any heat exchange with outdoors.

In modern building simulation software, this equation is solved for a much higher number of thermal nodes—a common practice is to have at least three nodes per wall, one for each surface and one in the middle. For multi-layered construction elements, the number of nodes is even higher. The computation time being polynomial in the number of nodes, a reasonable balance between accuracy and speed must be struck.

Our goal here is not to model the energy performance of a real office room, but rather to compare the energy performance of a simplified room under two different control strategies. As such, absolute accuracy is not required as much as relative accuracy. Adding more nodes to the model would increase its absolute accuracy but is unlikely to shed more light on the relative performance of one control strategy over another. A two-node model is, therefore, satisfactory for our needs.

For a thorough discussion of building simulation, we recommend Clarke (2001). Ferguson (1990) discusses also, in her thesis, the pros and cons of different thermal node placement strategies.

3.2.5 Solar gains model

The Fraunhofer Institute for Solar Energy (ISE) has carried out calculations (Kuhn, 2006a,b) of the total solar energy transmittance, or *g-value*, for different blinds used in this project. The calculations yield the g-values as a function of the source's azimuth and elevation (which can be negative for ground reflections) and of the slat angle. This data was used in the simulation to calculate the free gains due to solar gains.

The source's azimuth was discretized between $-80°$ and $+80°$ in steps of $20°$. Its elevation was discretized between $-85°$ and $+85°$ in steps of $5°$. The slat angle was discretized between $0°$ (horizontal slats) and $75°$ (closed slats) in steps of $15°$.

The ISE produced one file per slat angle for each blinds. This data includes the glazing's g-value (75% for the simulated office room, 63% for the LESO glazings). File `epfl_genius_noval_ext_30.dat`, for example, contains a 35×9 matrix with the g-values for the Noval blinds (c.f. Figure 3.5) tilted at $30°$ in front of LESO's glazing. The data in this file is shown in Figure 3.17.

The Simbad model assumes the office room's windows to be Pilkington insulight 4/12/4 glazings (g-value 0.75). The model includes outdoor blinds, taken to be of the same model (and g-value) as the indoor Genius blinds fitted in LESO's office rooms.

3.3 Chapter summary

In this chapter we have seen in what office rooms and on what hardware the control algorithm will run. Two office rooms of the LESO building have been fitted with venetian blinds, whose sensors and actuators have been included in the existing EIB building management system. Each office room is fitted with three additional illuminance sensors.

The program that interfaces between the control program and the building management system runs on its own dedicated machine, independently of the control program. Its public interface can be implemented by a computer program simulating the dynamic behaviour of an office room. We have built such a computer model with Simbad and given all of its characteristics.

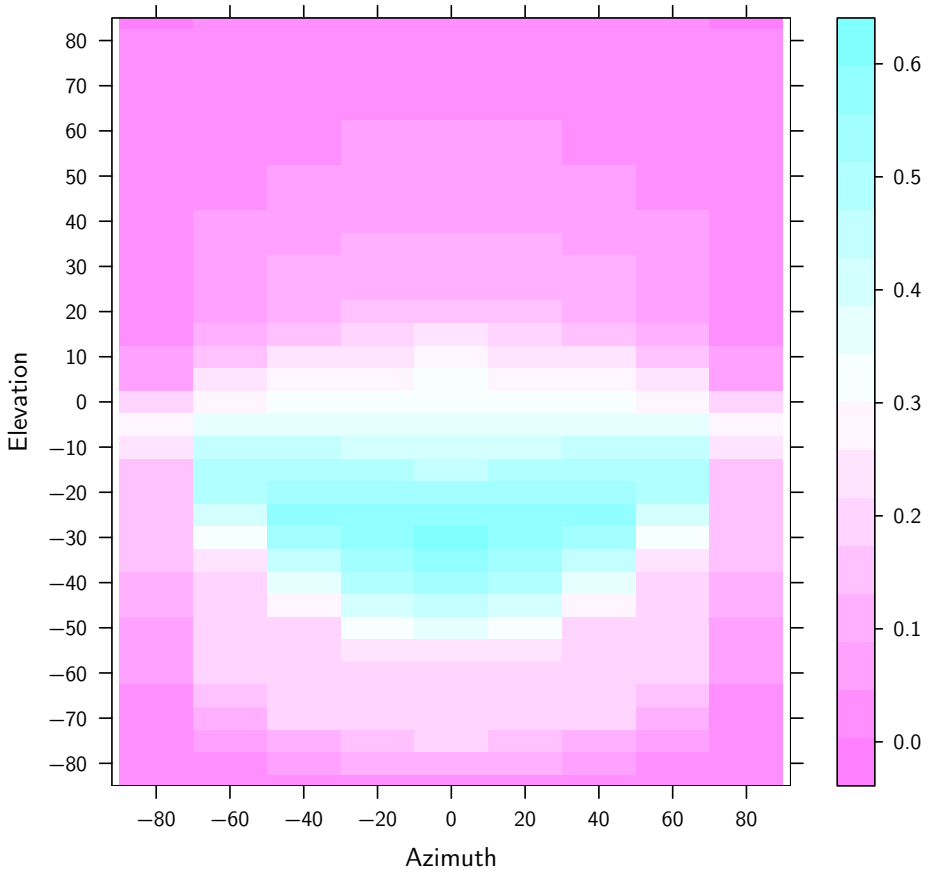


Figure 3.17: g-values for outdoor Noval venetian blinds with slats tilted 30°.

Our control program was written in Java and compiled against EIBSERVER's public interface given in section 6.3—not one line of EIBSERVER's code, or of any other building management system, or of SIMBAD is present in our controller's code. The ability to run our controller indifferently against a real building or a virtual one was not planned for at the beginning of the project—it arose spontaneously once it was realized that EIBSERVER's public interface could be separated from its implementation.

We find ourselves today in favourable conditions for the development of further control algorithms. A careful algorithm developer will first implement it in a prototyping language and run it against the computer simulation. After it has been shown that the implementation is correct and produces the expected results, the algorithm can be implemented in Java, assessed again against the simulation, and finally run against a real office room. This setup should dramatically shorten the development time for future control algorithms on the LESO building.

4 Daylighting model

In this chapter we will discuss the development of a simplified daylighting model suitable for embedded controllers.

The main purpose of this model is to provide a daylighting illuminance prediction method for the controller described in chapter 6. As we will see, our control algorithm evaluates the visual discomfort as a function of the illuminance observed at different points in the office room. A daylighting model is therefore needed if we are to find blinds' and electric lighting settings that minimize this visual discomfort.

A model that predicts daylighting illuminances for arbitrary zones and arbitrary shading devices is, however, also a valuable tool for many proposed control algorithms. For this reason, we document this model independently of the visual discomfort model described in the next chapter. The model described in this chapter could (and should) be easily implemented as an independent software module, capable of providing a daylighting model as a service to other software modules running on the same platform.

For these reasons, this chapter is mostly self-contained and can be read in isolation. Section 4.1 begins with some introductory remarks. In section 4.2 we discuss the Daylight Coefficients model, a model that would have been adopted for this project were it not for some critical shortcomings. In 4.3 we will first verify whether commonly used daylighting illuminance prediction methods are accurate enough for a daylighting controller. We will see that daylit scenes where direct sun is present cannot be accurately modeled with such models. In section 4.4 we will develop a linear daylighting model, valid for a given sun position. From this model we will deduce a more general one, which uses the history of past illuminances to assess the present illuminance. This model will be trained with a subset of simulated daylighting data, and validated on the remaining subset, in section 4.5. Finally, in section 4.6, we will build a working prototype controller set to maintain the horizontal workplane illuminance close to 500 lx and assess by simulation its performance over one year.

Glossary	
$E_{\text{in hor}}$	Indoor horizontal illuminance [lx]
$E_{\text{g hor}}$	Outdoor global horizontal illuminance [lx]
$E_{\text{g vert}}$	Outdoor vertical facade illuminance [lx]
$I_{\text{g hor}}$	Outdoor global horizontal irradiance [W/m^2]
$I_{\text{diff hor}}$	Outdoor diffuse horizontal irradiance [W/m^2]
I_{beam}	Direct beam irradiance [W/m^2]
$I_{\text{dir hor}}$	Direct horizontal irradiance [W/m^2]
DF	Daylight factor
ϵ	Perez sky clearness
$\tilde{\epsilon}$	Perez sky clearness category
θ	Sun elevation [°]
ϕ	Sun azimuth (0° is north, 90° is east) [°]
ϕ_{f}	Facade normal azimuth [°]

4.1 Model requirements

Many control algorithms that strive to maintain visual comfort while making an optimal use of shading devices (particularly if implemented on a low-powered, embedded controller) need a simple and reliable daylighting model of the space being controlled. How accurate does this daylighting model need to be? Keep in mind its purpose: to help our controller predict illuminances on selected room surfaces, and to help it achieve appropriate illuminances. The human eye will therefore be the yardstick which will define how precise we need to be. The relative errors due to this model (i.e. the ratio between error and real illuminance) must be smaller than illuminance differences the human eye can perceive. We have seen in section 2.7 that this ratio lies between 20–50%, but our target precision should be closer to 5–10%. But we have also seen that few daylighting modeling methods exist today that are both accurate and computationally cheap enough to be used on embedded hardware.

This chapter describes such a simple daylighting model intended for embedded daylight controllers. It provides a reasonably accurate prediction for daylighting illuminances for a controller that monitors the indoor illuminance on one hand, and the outdoor horizontal global and diffuse irradiance on the other hand. When asked to predict what illuminance would result from a given blinds' settings, the algorithm looks back in time and retrieves illuminance measurements that have been taken when the blinds were in a similar setting and the sun was close to its current position. This indoor illuminance is then modeled as a linear combination of the outdoor horizontal global and diffuse irradiances, and the current illuminance is predicted from this linear fit.

It is difficult, if not impossible, to develop, train and validate a daylighting model with real data. Experiments involving daylight are by nature impossible to repeat, and acquiring enough data takes far too long. We have therefore used RADIANCE-generated illuminances for different locations in a virtual office for each discretized venetian blinds' position and slat angle and for each minute of a year. This data was described in section 3.2.1, and the daylighting models evaluated in this chapter have, as far as possible, been validated against this data set.

4.2 Daylight coefficients

The Daylight Coefficients (DC) method, originally introduced by Tregenza and Waters (1983), is a method to completely characterize the illuminance response of an indoor point for an arbitrary sky luminance distribution.

The idea is to divide the sky vault in discrete, non-overlapping sectors small enough for the sky luminance to be considered constant across that sector. One usually follows Tregenza's division (Tregenza, 1987) into 145 non-overlapping circular zones between 11–12° in diameter. The dimensionless daylight coefficient D_i for sky patch i is then defined for an indoor point by:

$$D_i = \frac{\Delta E_i}{\Delta \Omega_i L_i} \tag{4.1}$$

where ΔE_i [lx] is the total (direct and reflected) illuminance at the point from sky patch i, L_i [cd/m^2] is the luminance of the patch, and $\Delta \Omega_i$ [sr] is its solid angle. The product $D_i \Delta \Omega_i$ is thus the proportionality coefficient between the illuminance at the point and the luminance of the patch (c.f. Figure 4.1). It measures directly how much the luminance of that sky patch

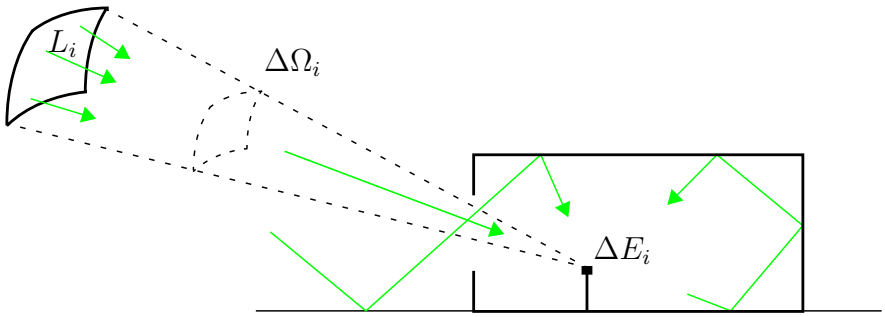


Figure 4.1: Daylight Coefficients, from Mardaljevic (1999). The indoor illuminance ΔE_i is the total (including reflected) illuminance from sky patch i, whose luminance is L_i and whose solid angle is $\Delta\Omega_i$.

contributes to the total illuminance at the point, including outdoor and indoor reflections.

Building a daylighting model based on DCs for an embedded controller is very attractive. It involves computing once and for all a set of DCs for each point of interest in the controlled zone, and for each possible blinds' positions and slat angles. DCs are, by definition, independent of the building's orientation (for identical outdoor obstacles) or of the sky vault's luminance distribution.

If there are more than one indoor point whose illuminance we want to model, let $E_1, E_2, \ldots, E_m$ be these m indoor illuminances to be modeled, and $\Delta\Omega_1 L_1, \Delta\Omega_2 L_2, \ldots, \Delta\Omega_n L_n$ the n products between solid angle and luminance for the n sky patches. If D_{ij} is the daylight coefficient for the ith indoor point and the jth sky patch, then:

$$\begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_m \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & \ldots & D_{1n} \\ D_{21} & D_{22} & D_{23} & \ldots & D_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ D_{m1} & D_{m2} & D_{m3} & \ldots & D_{mn} \end{bmatrix} \times \begin{bmatrix} \Delta\Omega_1 L_1 \\ \Delta\Omega_2 L_2 \\ \vdots \\ \Delta\Omega_n L_n \end{bmatrix} \tag{4.2}$$

Daylighting models based on daylight coefficients are computationally cheap, and have been shown to be accurate enough to compete with ray-tracing methods, provided the direct sun be treated with special care (Mardaljevic, 1999; Reinhart, 2001). The DC method assumes the luminance variation across a sky patch to be negligible enough to be considered constant, but this will be untrue for the patch in which the sun is. Smearing the sun across a 11° wide patch will introduce inaccuracies, which is why these authors recommend separately calculating the illuminance contribution from direct sunlight. Reinhart, in particular, proposes to compute a separate set of daylight coefficients for the sun's direct contribution since a typical Tregenza patch is large enough to contain 631 solar discs.

Reinhart has validated the DC method, as implemented in the DAYSIM software (Reinhart, 2006), against other methods that simulate the hourly illuminance over a year. DAYSIM was found to be the fastest of all (1.5 h on a Pentium Pro 200 MHz Linux PC) except for the daylight factor method (6 min), and was the most accurate except for the reference RADIANCE simulation (12 days).

Attractive as the Daylight Coefficients method is, we could not use it in this work. Our

project's vision was of a daylight controller that would require as little commissioning as possible, and we deem the computation of 145 daylight coefficients and direct light coefficients for each controlled office room, and for each possible position and slat angle of its venetian blinds, to be a complicated commission. Such a computation is beyond the scope of most installers, although it should not be beyond the skills of a good building designer.

Could DCs be measured in-situ in an office room? We have tried to answer this question in the early stages of this project by solving Equation (4.2) in the D_{ij} variables. We have no definitive answer. Early results suggest that it is difficult to obtain a set of sky luminance data that is neither singular nor "almost" singular, and standard least-squares solving techniques will fail. This further suggests the use of the Singular Value Decomposition technique, which we have not seriously attempted in this work.

We could not use DCs in this project, nor could we validate that method against our data since this data was produced with DAYSIM itself. Therefore we will not discuss this method further.

4.3 Daylight factor methods

In this section we will first test two simplified illuminance prediction methods, one of which is widely used, and compare their predictions with our simulated data. We will first consider our virtual south-facing office room in Brussels without any shading device.

4.3.1 Daylight factor

The daylight factor (DF) method assumes the indoor illuminance at a given point to be proportional to the outdoor, unobstructed, horizontal illuminance for a CIE overcast sky (CIE-1970; Moon and Spencer, 1942). This definition is valid, and predicts accurate daylight illuminances, only for CIE overcast skies whose distribution follows the Moon and Spencer (1942) distribution. For such a sky, the indoor illuminance $E_{\text{in hor}}$ is predicted by multiplying the daylight factor by the outdoor illuminance:

$$E_{\text{in hor}} = DF \times E_{\text{g hor}} \tag{4.3}$$

where $E_{\text{g hor}}$ is the outdoor global horizontal illuminance and DF is the daylight factor.

Our data does not directly provide the outdoor global horizontal illuminance but the outdoor global horizontal irradiance. Since the sky luminous efficacy is a constant 179 lm/W in RADIANCE[1] (Larson and Shakespeare, 2003, p. 357), this illuminance can be obtained by multiplying the irradiance by this efficacy. When validating any daylighting model with data generated with RADIANCE, we may therefore equivalently use the outdoor irradiance or outdoor illuminance.

Figure 4.2 shows the indoor horizontal illuminance against the global horizontal irradiance $I_{\text{g hor}}$, for different categories of Perez sky clearness. (The Perez sky clearness ϵ (Perez et al., 1993) is a measure of the sky's clarity and is equal to 1 for perfectly overcast skies and takes values up to 6 and more for perfectly clear skies. The Perez sky clearness category $\tilde{\epsilon}$ takes discrete values between 1 and 8 according to the value of ϵ and corresponds to increasingly clear skies.) For the cloudiest skies ($\tilde{\epsilon} = 1$), there is a good correlation between the horizontal

[1] This should not be confused with the usual daylighting value between 50–150 lm/W, depending on the type of sky.

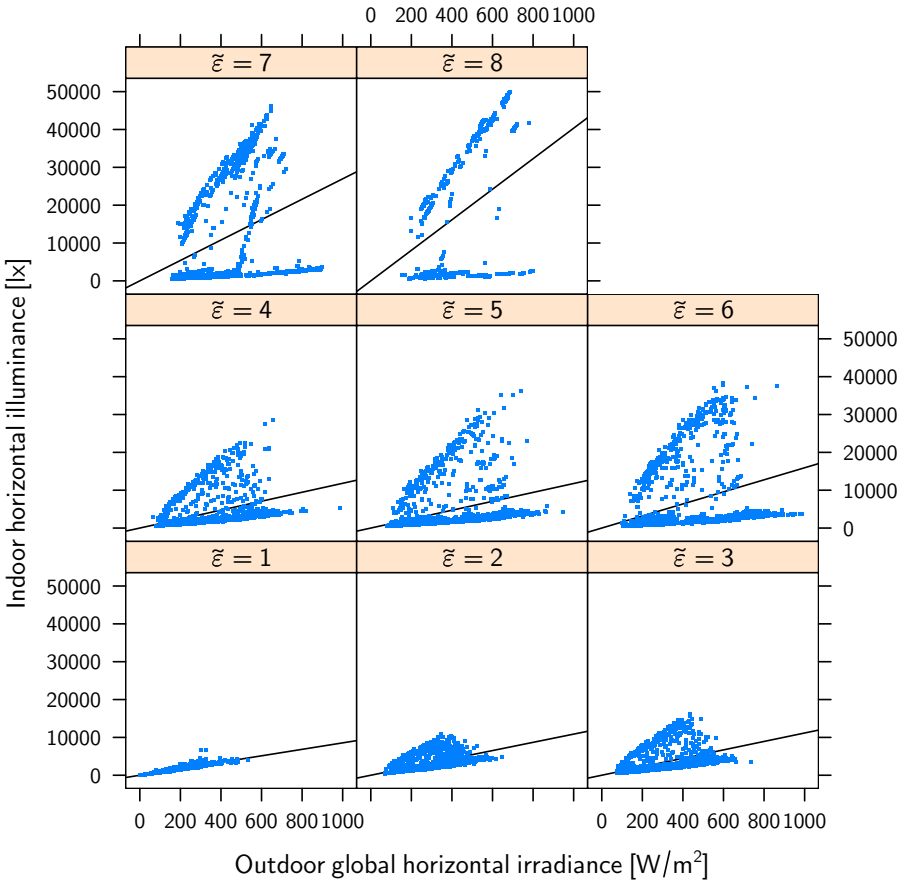


Figure 4.2: Indoor horizontal illuminance v. outdoor global horizontal irradiance conditioned on Perez's sky clearness categories. The best-fitting linear model is shown as a straight line in each panel. Not more than 1000 randomly chosen points are shown on each panel.

outdoor irradiance and the indoor illuminance, but the correlation breaks down for clearer skies.

In Figure 4.3 we plot the residuals against the fitted values of a daylight factor model, where the daylight factor is obtained by fitting a linear model between the indoor horizontal illuminance and the outdoor horizontal global irradiance. The high-illuminance points that depart the most from the red line tend to increase the linear coefficient and are responsible for the negative residuals at the other data points. They probably correspond to datapoints where the sun was present and visible from the point whose illuminance has been computed. This plot highlights the importance of taking the sun's position into account when predicting indoor illuminance, invalidating the daylight factor approach.

The daylight factor method is clearly unable to predict indoor illuminances for our test data other than under the most cloudy conditions. They were in fact never intended as a precise daylighting illuminance prediction method, but as a quick and easy way to verify the daylight performance of a building. They are now part of many building construction codes, but one

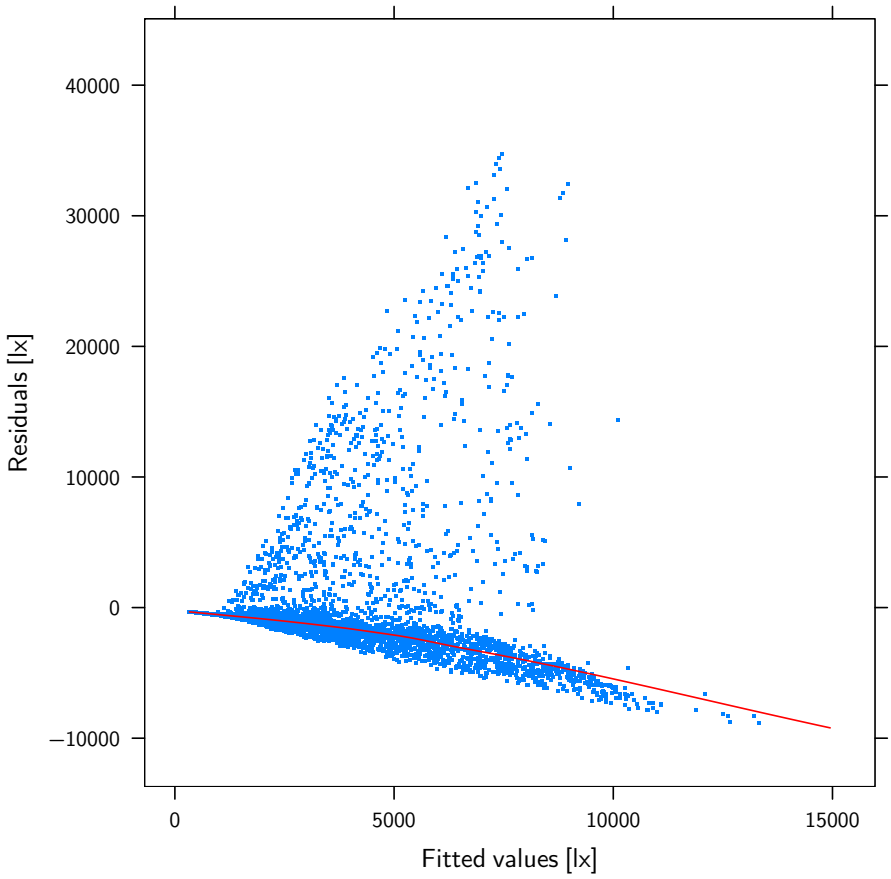


Figure 4.3: Residuals v. fitted values for the daylight factor model. The solid line is obtained by smoothing the scatterplot.

should keep their original purpose in mind before using them for illuminance prediction.

Indeed, Robinson and Stone (2004) have shown that isotropic sky models (or models without azimuthal dependence, such as the CIE overcast sky) cannot accurately predict the vertical irradiance on a window. Such models do not take the sun's contribution into account, and those errors will inevitably propagate on the calculation of the indoor illuminance.

4.3.2 Vertical irradiance

Guillemin (2003) has suggested that a better correlation than daylight factors might exist between the indoor illuminance and the outdoor vertical facade illuminance. He found experimentally that for the facades of the LESO building, the indoor illuminance $E_{\text{in hor}}$ could be modeled by:

$$E_{\text{in hor}} = a \exp(b \cdot \alpha) E_{\text{g vert}} \tag{4.4}$$

where α (between 0 and 1) is the fraction of the window not covered by a textile blind, $E_{\text{g vert}}$[lx] is the outdoor facade vertical illuminance, and a and b are model parameters to be fitted. For given blinds' settings, the indoor illuminance is therefore assumed to be proportional to the outdoor vertical illuminance:

$$E_{\text{in hor}} = \zeta \times E_{\text{g vert}} = \kappa \times I_{\text{g vert}} \tag{4.5}$$

using again the fact that in our simulated data $E_{\text{g vert}}$ is proportional to $I_{\text{g vert}}$, the outdoor vertical irradiance. Guillemin reports that the errors of this model have a standard deviation of 416 lx.

Assuming the diffuse irradiance to be isotropic, and neglecting ground reflections, that irradiance is the sum of half the total diffuse horizontal irradiance and of a direct component. It is given by:

$$I_{\text{g vert}} = I_{\text{diff hor}}/2 + I_{\text{beam}} \cos\theta \cos(\phi - \phi_f) \tag{4.6}$$

where θ is the sun's elevation, ϕ its azimuth, and ϕ_f the azimuth of the facade's normal, $I_{\text{diff hor}}$ [W/m^2] is the outdoor horizontal diffuse irradiance and I_{beam} [W/m^2] is the beam irradiance.

Figure 4.4 shows the horizontal illuminances plotted against $I_{\text{g vert}}$, conditioned on the sky clearness category as above. However, again, no clear correlation can be discerned for any but the cloudiest skies.

4.3.3 Fixed sun position

Figures 4.2 and 4.4 suggest that the sun's contribution cannot be ignored. Let us therefore redo these plots, selecting only those data points when the sun was not too far away from a given direction. We select sun positions within a 5° angular diameter circle[2] centered on elevation $\theta = 30°$ and azimuth $\phi = 190°$.

The two figures shown under in Figure 4.5, where different plotting symbols are used according to the Perez clearness category of the sky, are encouraging, especially when plotting

[2] A short implementation note is in order here. If $\theta_1, \phi_1, \theta_2, \phi_2$ are the respective elevations and azimuths of two solar positions, the angular distance between the two points is given by:

$$\Delta\sigma = \cos^{-1}(\sin\theta_1 \sin\theta_2 + \cos\theta_1 \cos\theta_2 \cos\Delta\phi) \tag{4.7}$$

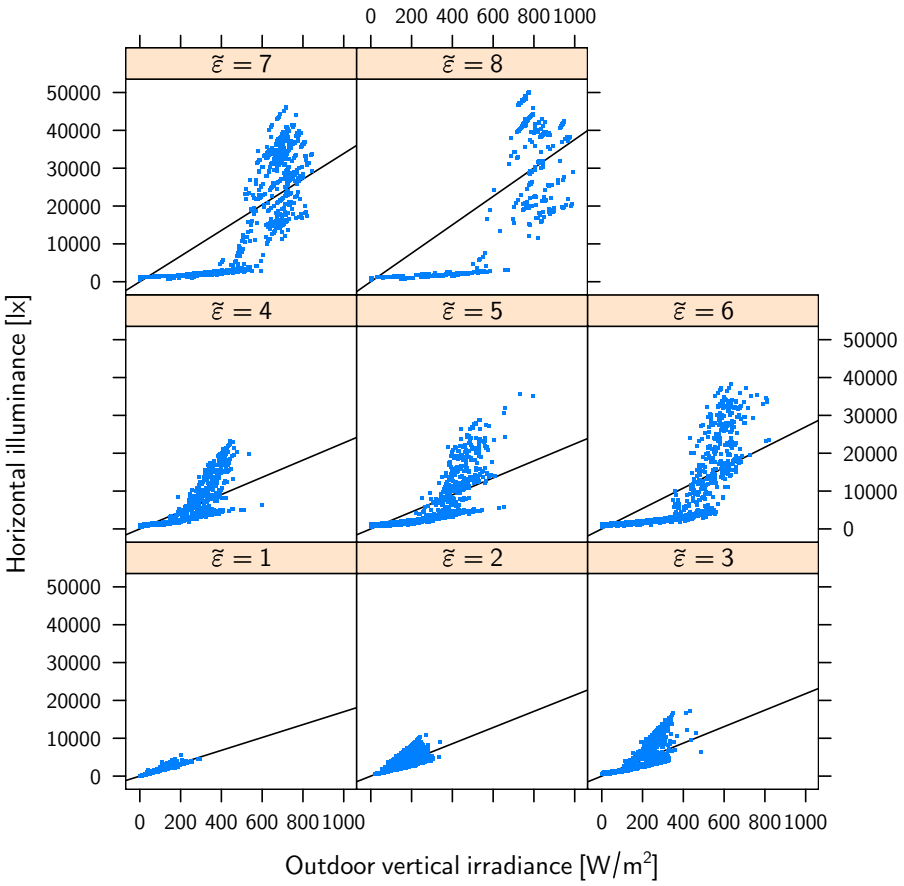
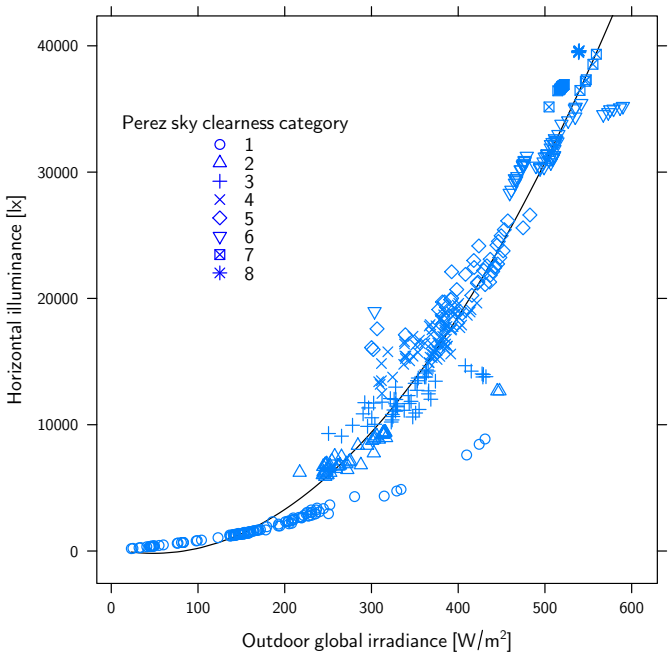
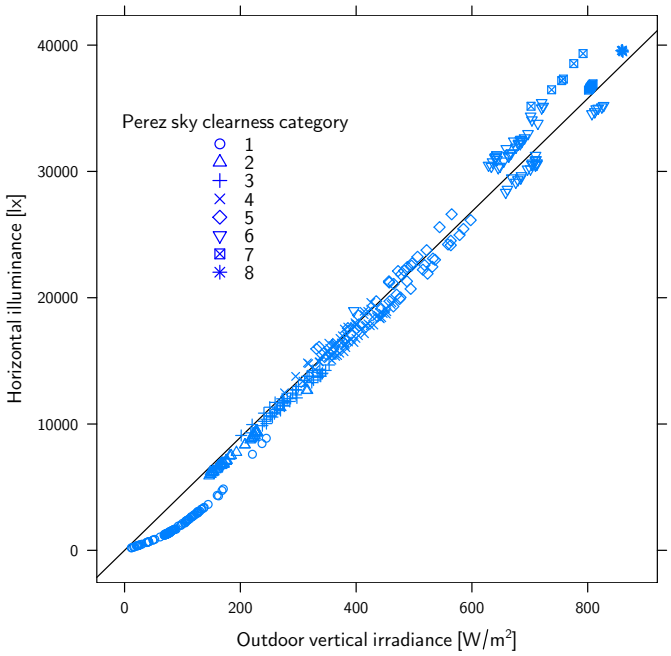


Figure 4.4: Indoor horizontal illuminance v. outdoor vertical irradiance conditioned on Perez's sky clearness categories. The best-fitting linear model is shown as a straight line in each panel. Not more than 1000 randomly chosen points are shown on each panel.



(a) Outdoor global irradiance



(b) Outdoor vertical irradiance

Figure 4.5: Indoor horizontal illuminance v. outdoor global and vertical irradiance ($\theta = 30°$, $\phi = 190°$), with a second degree polynomial fit for the global irradiance and a linear fit for the vertical irradiance.

	time	Eev	Eeg	Eed	rightEye	leftEye	horiz	clearness
468869	2006-11-22 14:28:30	85.6	130	123	3443	1912	1685	1.07
468870	2006-11-22 14:29:30	90.5	131	123	7295	2773	3013	1.08

Table 4.1: Data points 468869 and 468870. The column names correspond to the fields defined in Table 3.6 and to the variable names in the analysis code (Appendix C.1). The irradiances are given in [W/m^2], the illuminances in [lx].

against $I_{\mathrm{g\,vert}}$. But they also indicate a problem in our data for the most cloudy skies (clearness category $\tilde{\epsilon} = 1$), plotted as circles.

The RADIANCE extension program `gendaylit` is used by the DAYSIM program to generate the luminance distribution of the sky for each data point. This program implements the Perez All-Weather Sky Model (Perez et al., 1993) and takes as inputs the date, time, site coordinates, and direct and diffuse irradiance values.

From our data, consider the two contiguous data points 468869 and 468870 given in Table 4.1. Only one minute separates these two data points and the $I_{\mathrm{g\,hor}}$ and $I_{\mathrm{diff\,hor}}$ values are practically constant, but there is a sharp discontinuity in the modeled illuminance on all sensor positions.

The first row is almost a sky of clearness category 1 ($\epsilon < 1.065$), and it is conceivable that the internal algorithm of `gendaylit` did classify this sky as such. If this is true, then there could be a discontinuity in the luminance distributions given by the Perez model when moving from a sky of clearness category $\tilde{\epsilon} = 1$ to $\tilde{\epsilon} = 2$. This would explain the sharp illuminance increase, and the apparent anomaly of $\tilde{\epsilon} = 1$ skies in Figure 4.5.

When dealing with RADIANCE-generated data this model must therefore be fitted separately to $\tilde{\epsilon} = 1$ skies and to $\tilde{\epsilon} \neq 1$ ones. The result for a linear model against the outdoor vertical irradiance is shown on Figure 4.6. The correlation, in both cases, is now excellent.

4.4 Simplified daylighting model for a given sun position

In the previous section we found that for a given position of the sun (arbitrarily chosen as $\theta = 30°$ and $\phi = 190°$), the indoor illuminance could be reasonably well predicted as being proportional to the vertical facade irradiance. This irradiance was expressed as a linear combination of $I_{\mathrm{diff\,hor}}$ and I_{beam} in Equation 4.6.

The I_{beam} beam irradiance term of that equation can further be expanded to:

$$I_{\mathrm{beam}} = (I_{\mathrm{g\,hor}} - I_{\mathrm{diff\,hor}}) / \sin\theta \tag{4.9}$$

and $I_{\mathrm{g\,vert}}$ is therefore also a linear combination of $I_{\mathrm{g\,hor}}$ and $I_{\mathrm{diff\,hor}}$. Equation 4.5 can thus

where $\Delta\phi$ is the difference of the azimuths. This formula suffers however from rounding problems for small distances and another formula, known historically as the haversine (WIKI-GCD) formula, is preferred:

$$\Delta\sigma = 2\sin^{-1}\left(\sqrt{\sin^2\frac{\Delta\theta}{2} + \cos\theta_1\cos\theta_2\sin^2\frac{\Delta\phi}{2}}\right) \tag{4.8}$$

It is this latter formula that has been used throughout this work.

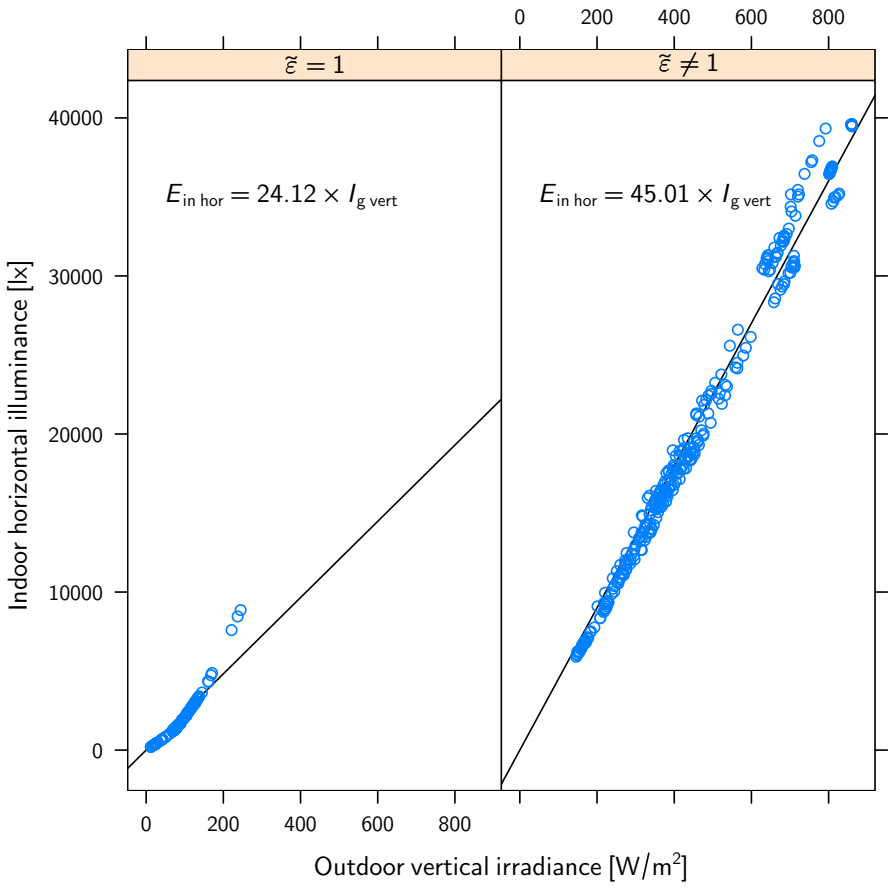


Figure 4.6: Linear fit of indoor horizontal illuminance against outdoor vertical irradiance, for $\tilde{\epsilon} = 1$ skies (left panel) and other skies (right panel).

	Estimate	Std. Error	t value	Pr(>|t|)
α	79.2165	0.1056	750.12	0.0000
β	−61.1011	0.2126	−287.45	0.0000

Table 4.2: Linear regression results on example data.

be generalized as:

$$E_{\text{in hor}} = \alpha \times I_{\text{g hor}} + \beta \times I_{\text{diff hor}} \tag{4.10}$$

So if Guillemin's hypothesis is true and $E_{\text{in hor}}$ is indeed equal to $\kappa \times I_{\text{g vert}}$ for a certain κ, then by identification $\alpha = -\kappa\frac{\cos\phi}{\tan\theta}$ and $\beta = \kappa(1/2 + \frac{\cos\phi}{\tan\theta})$. When we fitted a linear model to skies of clearness category other than 1, we found that $\kappa = 45.01$, so we expect:

$$E_{\text{in hor}} = 76.08 \times I_{\text{g hor}} - 53.78 \times I_{\text{diff hor}} \tag{4.11}$$

But the results of fitting the model of Equation 4.10 to our simulated data, given in Table 4.2, do not bear this out. The 95% confidence intervals for the two coefficients are:

$$E_{\text{in hor}} = (79.22 \pm 0.21) \times I_{\text{g hor}} - (61.10 \pm 0.42) \times I_{\text{diff hor}} \tag{4.12}$$

which is close to, but significantly different from, the values found in Equation (4.11). The $E_{\text{in hor}} = \kappa \times E_{\text{g vert}}$ hypothesis must therefore be rejected in favour of the more general expression in Equation (4.10).

Figure 4.7 shows the relative residuals when modeling the indoor illuminance as a linear combination of outdoor global and diffuse irradiance. The relative residuals tend to be higher for lower illuminances, but 98% of the points are within 10% relative error and 90% are within 5%. This model give a satisfactory fit and is valid for a given sun position. It is therefore on this model that our daylighting model will be built.

Indeed, early versions of this work were motivated by an attempt to model the indoor illuminance $E_{\text{in hor}}$ for a given scene configuration[3] as a linear combination of outdoor direct horizontal irradiance $I_{\text{dir hor}}$ and diffuse irradiance $I_{\text{diff hor}}$:

$$E_{\text{in hor}} = \gamma \times I_{\text{dir hor}} + \delta \times I_{\text{diff hor}} \tag{4.13}$$

This equation assumes, perhaps naively, that all daylight in an office room comes from only two sources—the sun and the rest of the sky vault. The latter is reduced to a single light source, and this model has in jest been referred to as a Daylight Coefficients model with just one sky patch instead of 145. As we shall see, the errors introduced by this simplifying assumption turn out to be negligible, provided the constancy of the scene's configuration is respected. In fact, this relation is physically exact if the sky vault's *relative* luminance distribution is constant (i.e., the luminance of each point is doubled if the global diffuse irradiance is doubled).

Many solarimeters measure $I_{\text{dir hor}}$ or $I_{\text{diff hor}}$, but seldom both. $I_{\text{dir hor}}$ is a linear combination of $I_{\text{g hor}}$ and $I_{\text{diff hor}}$, which is why we may equivalently use the convention proposed

[3] We define a "scene configuration" as a scene where the positions of every object is fixed. In particular, any blinds' settings are fixed, as is the sun's position. The outdoor direct and diffuse irradiances are free variables.

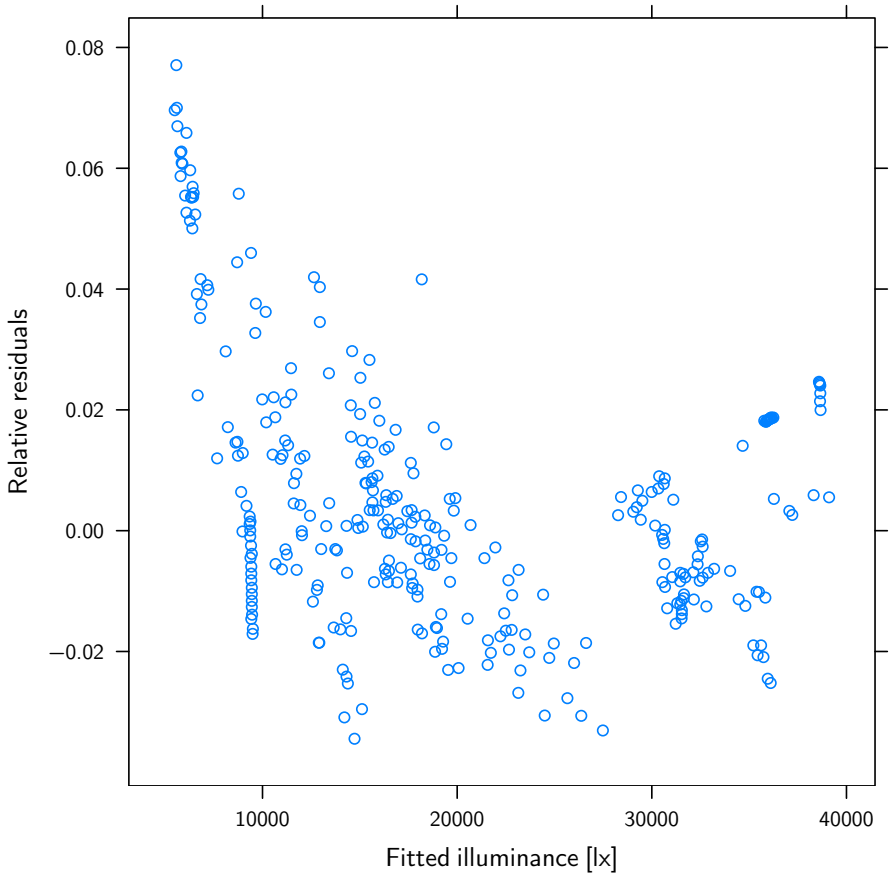


Figure 4.7: Relative residuals v. fitted values

above:

$$E_{\text{in hor}} = \alpha \times I_{\text{g hor}} + \beta \times I_{\text{diff hor}} \tag{4.14}$$

This choice is arbitrary. We made it because our BF3 solarimeter (c.f. Figure 3.10), like many solarimeters, measures global and diffuse irradiance.

What we are actually doing is fitting an overdetermined linear model. If we make N observations for the same or similar scene configurations but different irradiance values, and introduce an ϵ term to account for measurement and systematic errors, we can then write:

$$\begin{aligned} E_{\text{in hor}1} &= \alpha\, I_{\text{g hor}_1} + \beta\, I_{\text{diff hor}1} + \epsilon_1 \\ E_{\text{in hor}2} &= \alpha\, I_{\text{g hor}_2} + \beta\, I_{\text{diff hor}2} + \epsilon_2 \\ &\vdots \\ E_{\text{in hor}N} &= \alpha\, I_{\text{g hor}_N} + \beta\, I_{\text{diff hor}N} + \epsilon_N \end{aligned}$$

This can more conveniently be written in matrix form:

$$\boldsymbol{E} = \mathsf{I} \times \boldsymbol{\psi} + \boldsymbol{\epsilon}$$

where $\boldsymbol{E}$ is the vector of illuminance observations, I is a $n \times 2$ matrix with the $I_{\text{g hor}}$ and $I_{\text{diff hor}}$ irradiance data, $\boldsymbol{\psi}$ is a 2-element vector with the fitted parameters α and β, and $\boldsymbol{\epsilon}$ is the vector of errors.

Our goal is to find the 2-element vector $\hat{\boldsymbol{\psi}}$ which minimizes the norm of $\boldsymbol{\epsilon}$ (least-squares solution). Assuming that the 2×2 matrix $\mathsf{E}^{\mathrm{T}}\mathsf{E}$ is of full rank and therefore invertible, and that the errors are normal, the solution is given by:

$$\hat{\boldsymbol{\psi}} = (\mathsf{I}^{\mathrm{T}}\mathsf{I})^{-1}\mathsf{I}^{\mathrm{T}}\boldsymbol{E} \tag{4.15}$$

which is one way that this model could be implemented in a controller. For efficiency reasons our controller will implement this model slightly differently, as described in chapter 6.

4.5 Validation by simulation

In this section we use our simulated data to validate the model proposed above. It was shown to yield satisfactory results for all kinds of skies with the sun in an arbitrary position. We must now verify if it works for other directions of the sun and for different sets of data.

In section 4.5.1 we will train our model with daylighting data from the first half of the year, and use that data to predict the daylighting illuminance during the seconf half of the year. In section 4.5.2 we will predict the daylighting illuminance for the whole year, using only data that is at least one week old. Finally in section 4.5.3 we will repeat the whole year illuminance prediction but for two different scene configurations: first with a west-oriented virtual office room with retracted blinds, second with a south-oriented office room with venetian blinds closed and slats in a horizontal position.

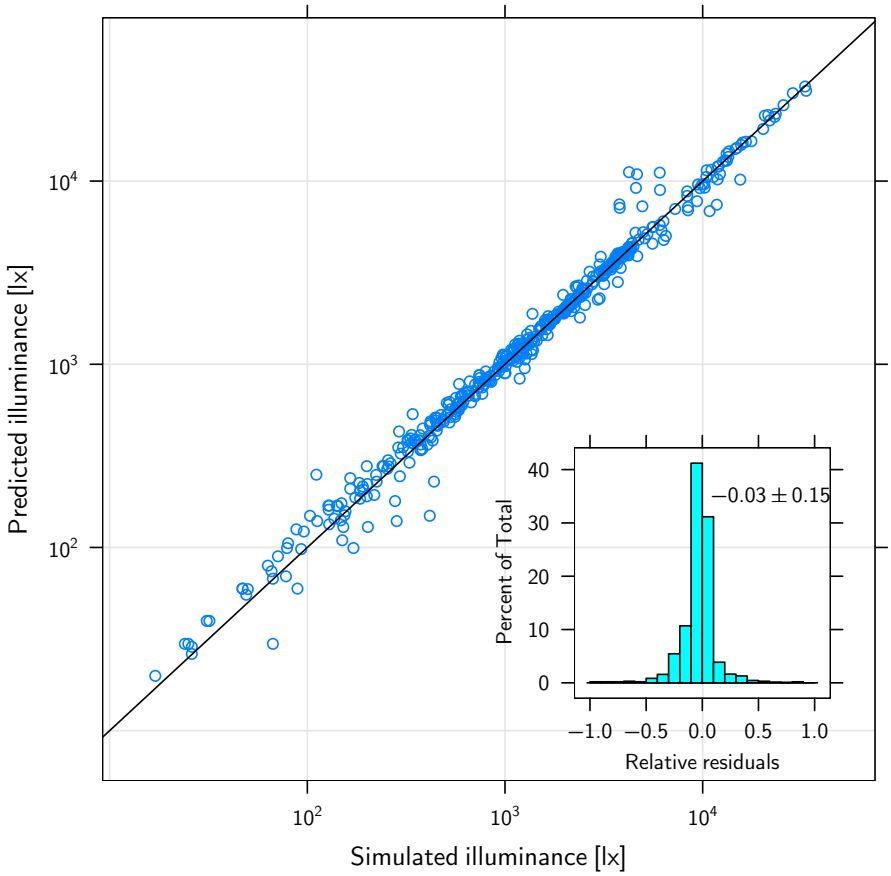


Figure 4.8: Predicted July–December hourly indoor horizontal illuminances, using only January–June data. 1000 randomly chosen points are shown out of the 4417 simulated points. The scale is logarithmic. The line's slope is 1 and its intercept 0. The relative residuals are histogrammed and their mean and standard deviation are given.

4.5.1 Half-year training data

First we will predict the hourly illuminances during the months of July to December, using only the January to June data for training. The virtual office room is south-oriented and its venetian blinds are retracted. At each time step, we select from the first half year the data points with a similar sun position and a similar sky clearness category. From these points, we derive α and β for that timestep, and predict its indoor horizontal illuminance.

Figure 4.8 shows the predicted v. simulated horizontal illuminance. The correlation is excellent ($R^2 = 0.98$), and proves that our model is sound.

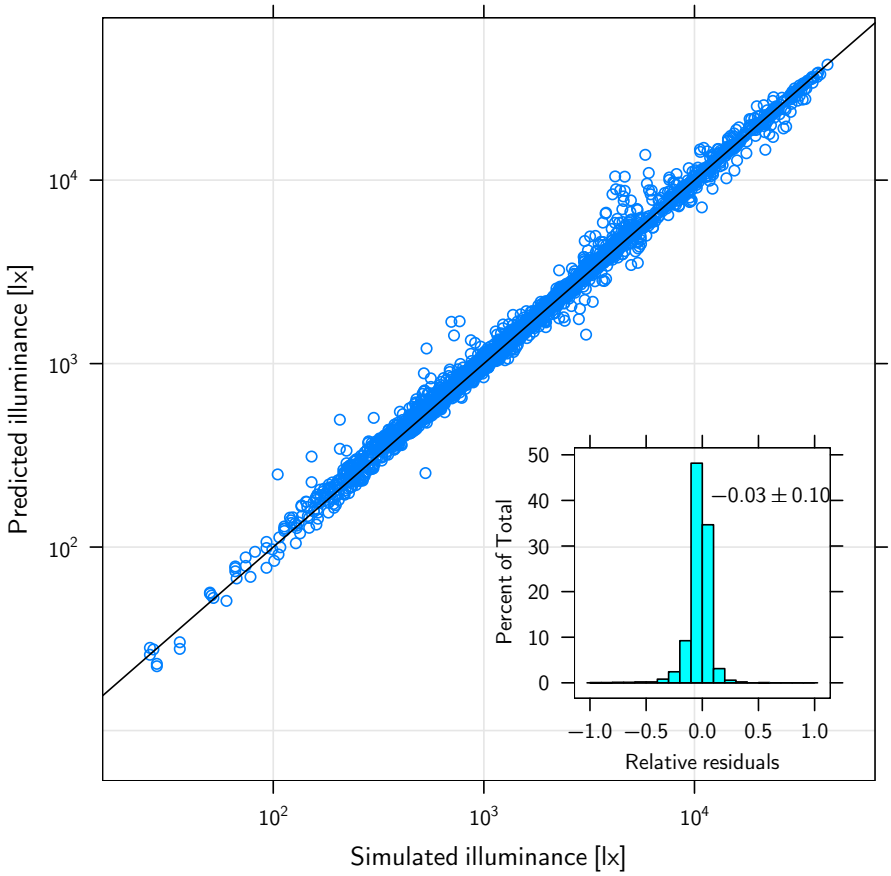


Figure 4.9: Predicted v. simulated illuminances during progressive learning. The scale is logarithmic. The line's slope is 1 and its intercept 0. The relative residuals are histogrammed and their mean and standard deviation are given.

4.5.2 Progressive learning

How fast does the model learn? If the model is to be used in a daylighting controller at all it must learn reasonably fast. A controller that has to wait half a year before making precise predictions is not useful.

Therefore, here we will let the model predict the indoor horizontal illuminance every hour for our south-oriented virtual office room with retracted blinds. At each timestep we use training data older than at least a week. On 1 February, for example, only the data up to 21 January may be used to predict the illuminance. We should not use data younger than that, because it would be too easy for the model to use training data from, say, the day before—the behaviour of an office room with respect to daylight does not change that much from one day to the other: the sun's trajectory will be similar, and chances are good that the weather will be similar too.

Figure 4.9 shows the predicted against simulated horizontal illuminances on a logarithmic scale during the simulated year. The correlation is excellent ($R^2 = 0.99$). Figure 4.10 shows

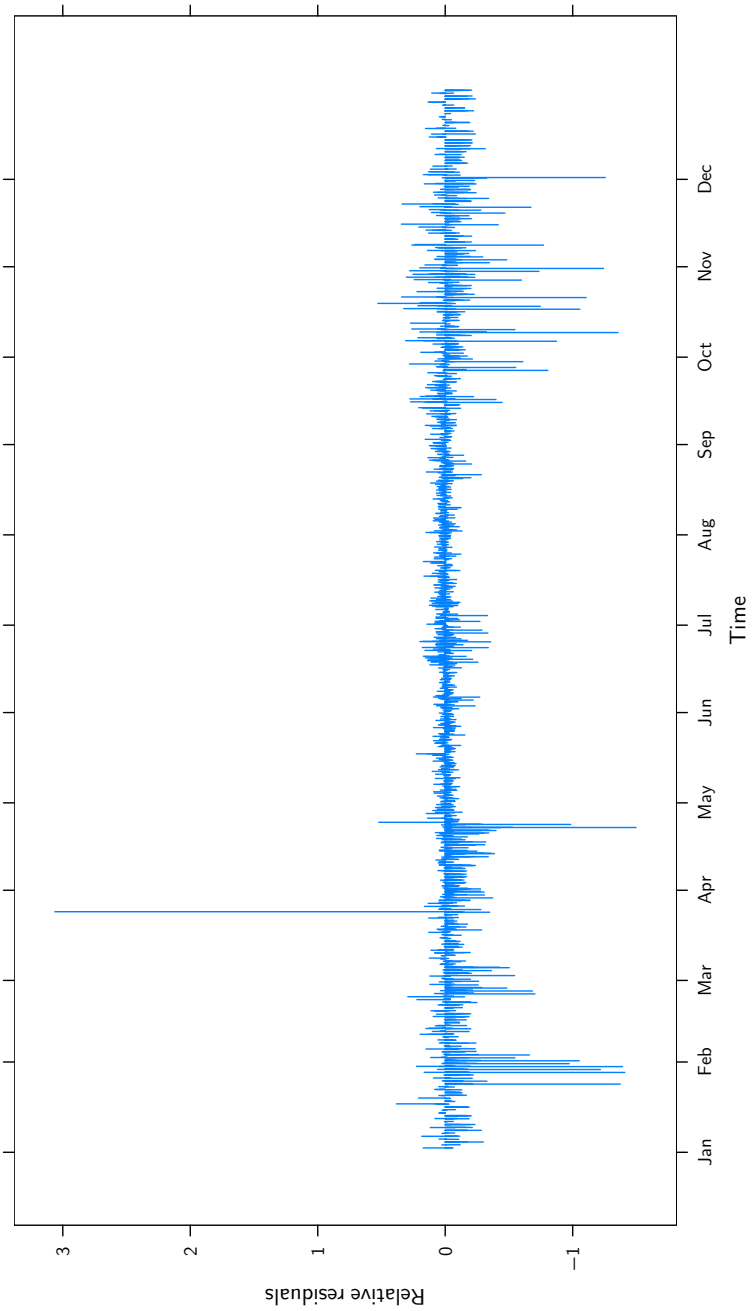


Figure 4.10: Relative residuals during progressive learning.

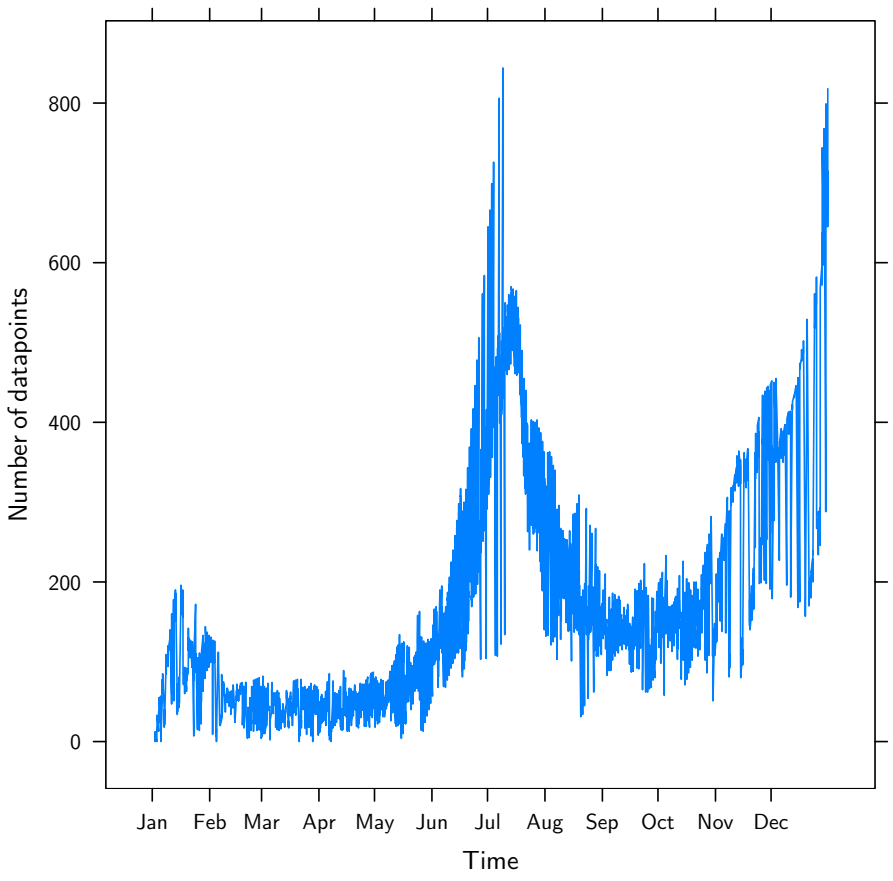


Figure 4.11: Number of points retained for illuminance modeling over the year during progressive learning.

how the relative residuals evolve over time. One would naively expect them to be greatest in amplitude during the first months and then gradually decrease as the algorithm learns, but instead of that they do not really improve over time. They are worst in spring and autumn, and very good in summer and winter.

A possible explanation for this is given by Figure 4.11, where the number of datapoints used at each timestep is plotted against time. The angular distance between two consecutive daily solar trajectories in the sky is not constant over the year. They are closer in summer and in winter than in spring and autumn (see Figure 4.12). Therefore, when predicting the illuminance on a spring day, the model has more difficulties finding previous similar scene configurations than in summer. In summer and winter, the sun's consecutive trajectories in the sky lie very close to one another and the model is better at finding previous similar situations.

Note, however, that these observations hold only for the first year of operation. After a full year of learning, this artefact will not exist any longer and the model will be even more accurate.

4.5.3 West-facing facade and venetian blinds

In this section we will carry out the same validation as in the preceding one, but for two different situations. First for a west-facing facade orientation with retracted blinds, and then for a south-facing facade protected with fully lowered venetian blinds and slats in a horizontal position.

The predicted v. simulated illuminance scatterplots are given in Figure 4.13, where they are also compared with the plot already given in Figure 4.9. In all cases, the correlation is excellent and the predicted values agree well with the real ones. The case with venetian blinds performs visually the least well, probably because of the increased complexity.

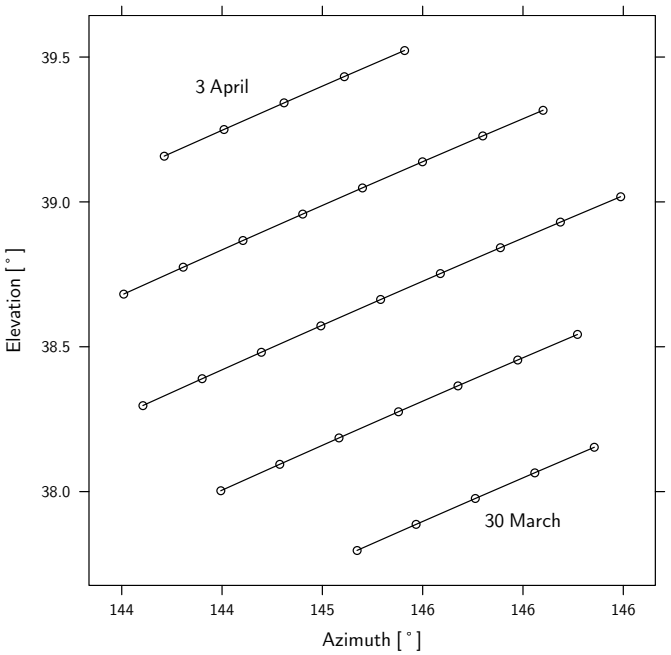
Histograms of the three relative residuals are given in Figure 4.14. There is no apparent difference between the three cases, the model being quite able to predict the horizontal illuminance in all situations.

4.6 Implementation in a daylighting controller and test on a virtual office room

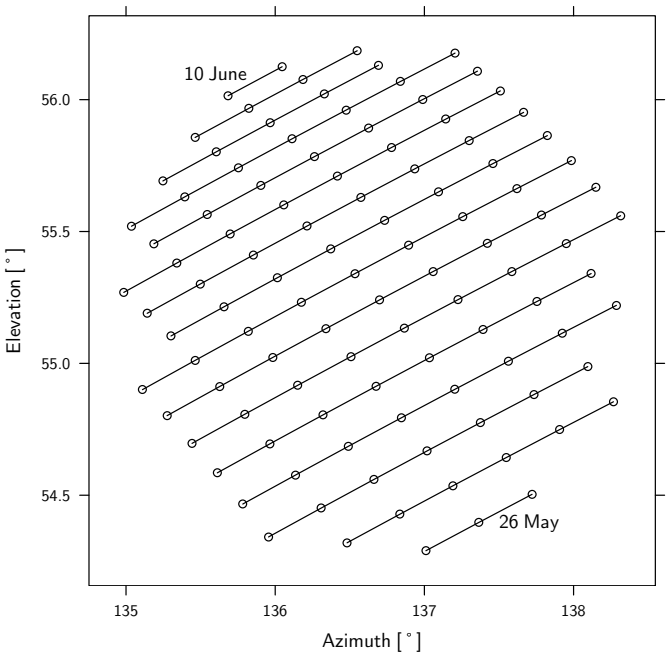
We have developed a prototype daylighting controller, that implements the ideas presented in this chapter. Its purpose was to prove that the daylighting model presented in this chapter could be used by a simple controller, and to help in optimizing the implementation of this model. In this section we succintly describe the elements of this prototype, many of which were subsequently reused for the controller described in this work.

An element of this prototype is a data acquisition (DAQ) module, programmed to run only on weekends and to stop immediately if a user is detected. After sunrise, it iterates sequentially through the blinds' positions and slat angles. The positions are discretized in steps of 20% of the total window opening, and the slat angles in steps of 10%. The illuminance sensors in the office rooms write out their measured values once about every 40 s, so the DAQ module waits for two minutes before recording the monitored illuminance. It stops at sunset.

On each cycle, the blinds thus move through 66 different positions (6 positions $\times$ 11 slat angles). A complete cycle, allowing for two minutes at each step, takes thus about two hours



(a) 1 April



(b) 1 June

Figure 4.12: Sun positions at one-minute intervals. Sun positions closer than 1° angular distance from the 10 a.m. sun on the given dates are shown. Each trajectory corresponds to solar courses on consecutive days. Notice how much closer to each others consecutive solar courses are in early June compared to early April.

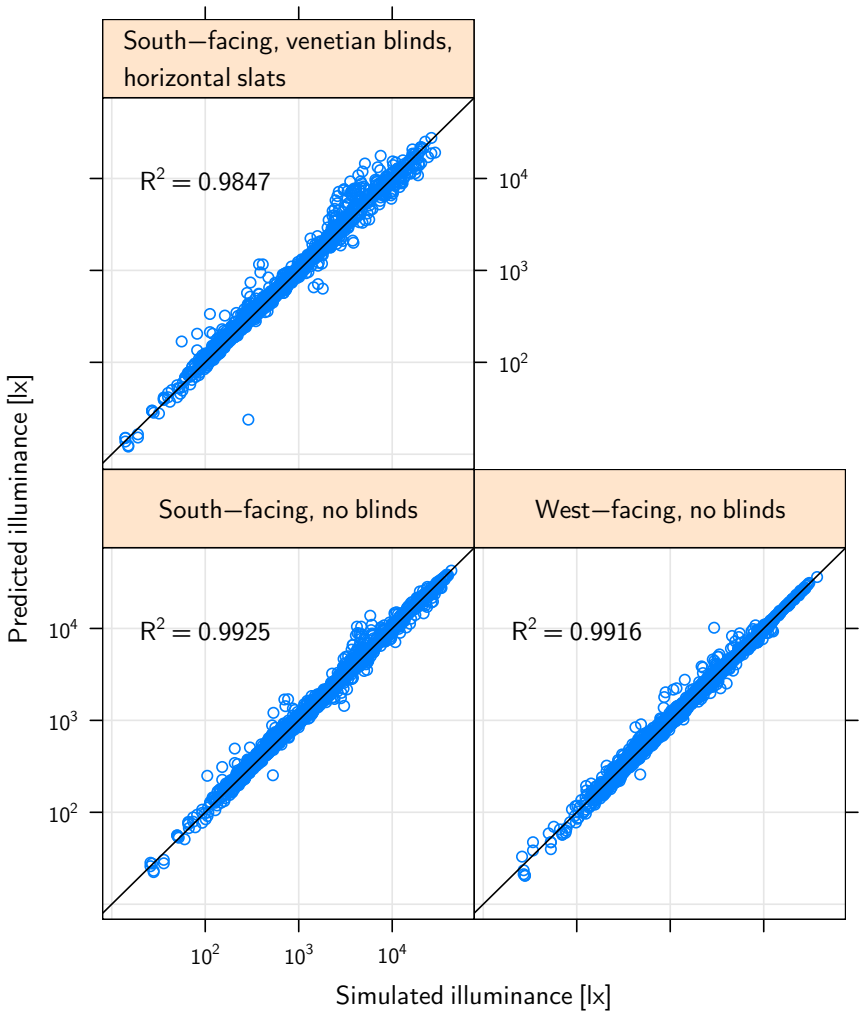


Figure 4.13: Predicted v. real illuminances during progressive learning for three different cases. A line of slope 1 and intercept 0 is drawn in each panel.

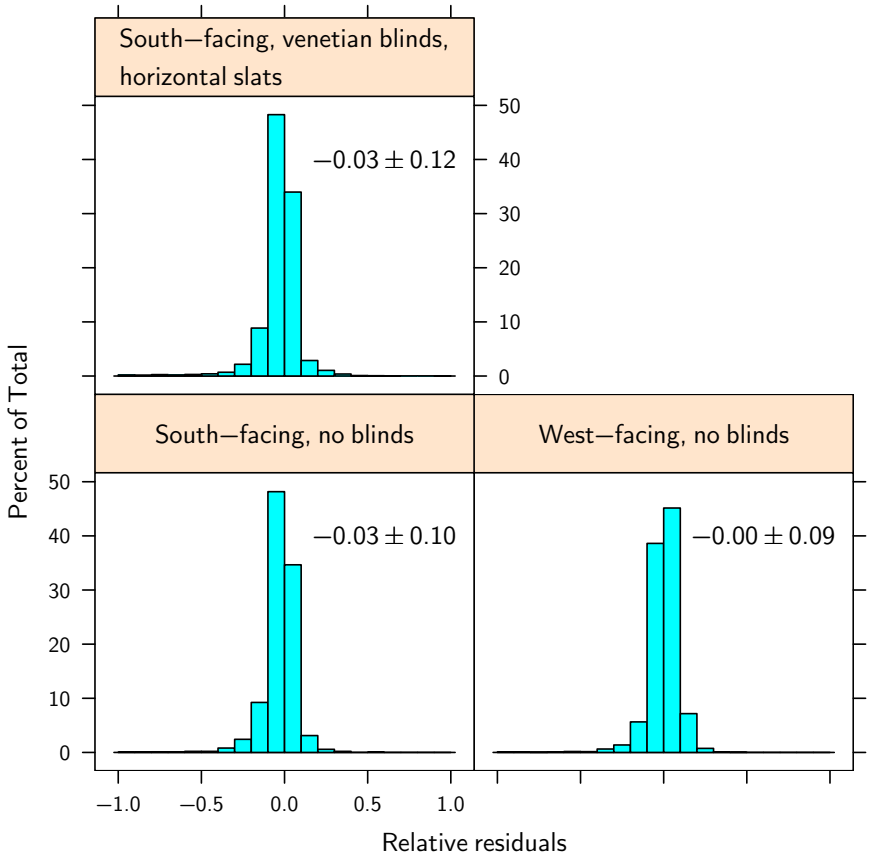


Figure 4.14: Relative residuals for the three simulations. The mean and sample standard deviation are given.

realtime. The initial position and slat angle at each cycle are chosen randomly to prevent bias.

The results are stored in a text file. On each line we record the date, the time, the outdoor global and diffuse irradiance, the blinds' position and slat angle and the illuminances.

Another module of the prototype is responsible for predicting the horizontal illuminance for arbitrary blinds' settings and arbitrary outdoor global and diffuse irradiances. It does this by following the method presented in this chapter, with some optimizations that will be described in chapter 6.

At any timestep, the prototype identifies its model parameters from the data it was allowed to collect during the weekends up to that timestep. It begins the year with an empty data file, and begins collecting data on Sunday 1 January 2006.

A third module of the prototype is the optimizer. It is responsible for sending actual commands to the blinds and to the electric lighting. Every five minutes realtime, it models the illuminance of the office and explores the different blinds' settings and electric lighting power, seeking to minimize a given cost function. If a new situation can be found, whose cost function is significantly lower than that of the current situation, then the new settings are applied to the blinds and the electric lighting. A real controller should not annoy a user with too frequent blinds movements, so we limit the number of blinds movements to not more than one per 15 minutes.

Our prescription to the prototype controller will be to maintain a 500 lx horizontal workplane illuminance throughout the year, even at night (with electric lighting). Electric lighting is allowed because we need to detect those situations when the controller was unable to obtain 500 lx with daylight alone, e.g. during completely overcast skies in the late afternoon. By allowing electric lighting, but strongly biasing against it, we can identify when the controller used daylighting alone to obtain 500 lx.

The cost function will include a smooth, derivable function of $E_{\text{in hor}}$, the horizontal workplane illuminance, with a minimum at 500 lx. Our suggestion[4] is $\left(\frac{E_{\text{in hor}}-500}{500}\right)^2$. We want the controller to favour daylighting over electrical lighting, so we add a small arbitrary term proportional to P, the fraction between 0 and 1 of maximum power applied to the lighting fixture. P must be multiplied by a weight, that should be large enough to prevent the controller from using artificial lighting when daylighting is enough but small enough that the resulting illuminance at night remains close to 500 lx (if too large, the controller will never switch the electrical lighting on). A weight of 0.1 satisfies these conditions. The complete cost function is:

$$f(E_{\text{in hor}}, P) = \left(\frac{E_{\text{in hor}} - 500}{500}\right)^2 + P \times 0.1 \tag{4.16}$$

We have let this prototype run against the same virtual office described in section 3.2, and the resulting illuminance distribution on weekdays, when the controller believed no artificial illuminance was needed to maintain 500 lx, is given in Figure 4.15. The distribution is clearly centered around 500 lx, but is this attributable to the controller or is it the natural illuminance distribution for an arbitrary blinds' settings?

To answer this question, we give in Figure 4.16 the same illuminance distribution as in Figure 4.15, but plotted in a logarithmic scale and compared with the illuminance distributions over the complete year for the same points in time in the same office room with, in the

[4] In the final controller, a more complicated cost function will be given that takes into account the user's preferences.

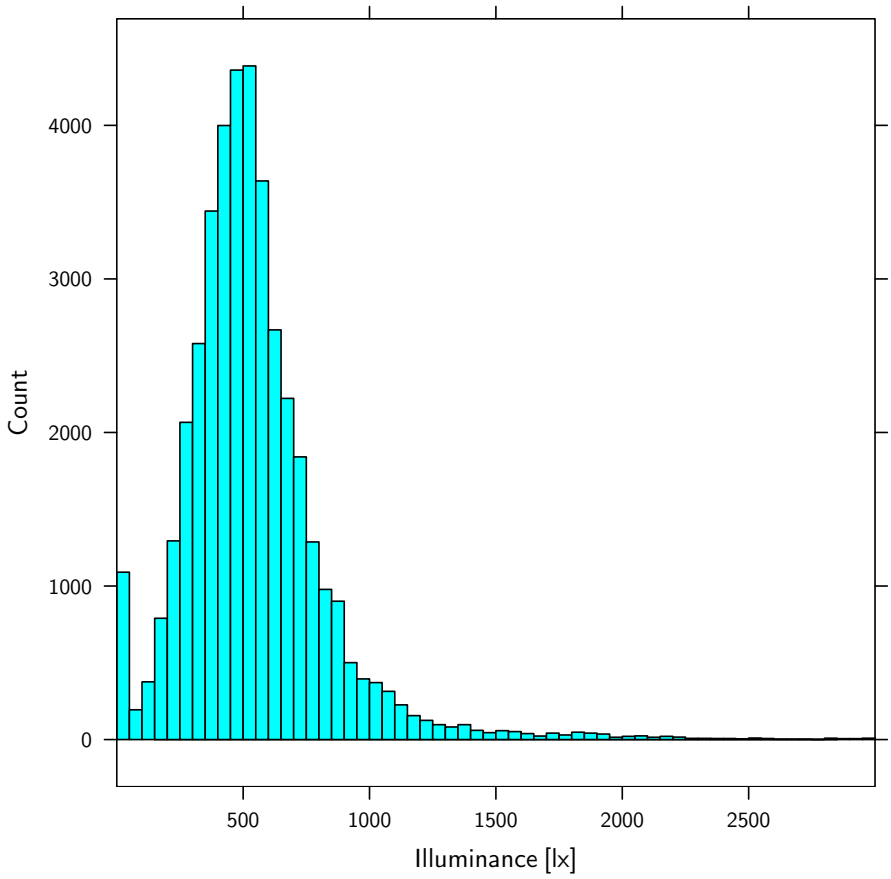


Figure 4.15: Illuminance distribution with daylighting controller when daylighting alone was thought by the controller sufficient to maintain a 500 lx workplane illuminance.

first panel, venetian blinds completely retracted, and in the second panel, venetian blinds completely down with slats in a horizontal position.

The controller has obviously improved the illuminance distribution. The latter is much narrower and clearly centered on 500 lx, and the two top panels are proof that this is not a naturally occuring condition. We will give an explanation for the observed low illuminance values towards the end of this section.

The sample standard deviation of the logarithm of the illuminance without a controller present are 0.430 and 0.480 respectively. With the controller present, the sample standard deviation narrows down to 0.219. Assuming for simplicity that the parent population is normally distributed (it almost certainly is not—a Shapiro-Wilk test for normality on the sample population gives a p-value smaller than 10^{-20}), and remembering that we took the logarithm of the illuminance values, this means that the controller keeps the horizontal workplane illuminance on average at 503 lx, with 95% confidence intervals between 187–1351 lx. This is satisfactory for most practical applications.

The last point we would like to investigate is the bin with low illuminance values on the lower panel of Figure 4.16. It corresponds to situations where the controller was mistaken, probably because of insufficient training data. To test this hypothesis, we show in Figure 4.17 how the daylight illuminance distribution varied over the year. The low illuminance values correspond exclusively to situations in February and March where the controller was mistaken. This is expected, because in early months the controller has not yet acquired enough training data, especially in spring when the daily sun courses are far apart. They are lacking in January because this was a month during which the controller often did not even have enough data to model the daylight at all, in which case it was programmed to do nothing. The controller's performance for the rest of the year is satisfactory.

4.7 Chapter summary

In this chapter we have examined classical daylight prediction methods and found that each had its strengths, but none fulfilled all the requirements for a self-commissioning daylighting controller. The most promising approach, that of the Daylight Coefficients method, required too much up-front computation to be succesfully used on typical installations.

We have proposed a learning algorithm, whereby a software module in the controller records indoor illuminances for different outdoor global and diffuse (or equivalently, global and direct) irradiances and for different sun positions and blinds' positions and slat angles. When modeling the illuminances for a new scene configuration, the algorithm looks up data recorded in similar scenes and models the illuminances as a linear combination of global and diffuse outdoor irradiances.

We have seen that this model performed satisfactorily against our synthetic data. The R^2 correlation between predicted and simulated illuminances, in three different test cases, was 0.98, 0.99 and 0.99. We have built a prototype controller using this model and tested it against our virtual office during a year of simulated operation. When daylighting alone permitted it, the workplane illuminance was kept to an average 503 lx, 95% of the time between 187–1351 lx. We have also described some of the software modules that will ultimately be responsible for modeling daylight illuminances for the controller described in this work.

We have now in principle a daylighting controller capable of keeping the indoor illuminance at any prescribed value. What this value should be will be the subject of the next chapter.

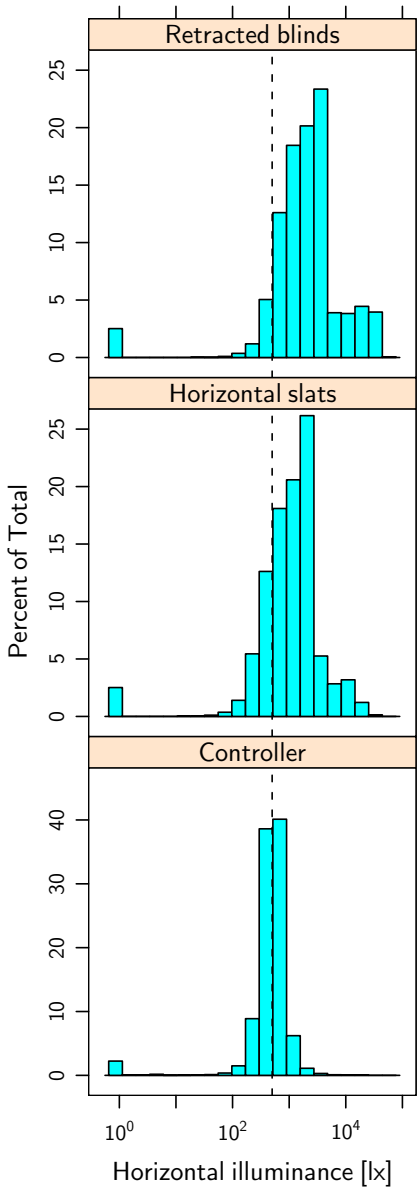


Figure 4.16: Illuminance distribution with or without daylighting controller. From the top, the panels show the situation with venetian blinds completely retracted, completely down but slats in a horizontal position, and with the prototype controller. The dashed line corresponds to 500 lx. Notice the change of vertical scale on the lower panel.

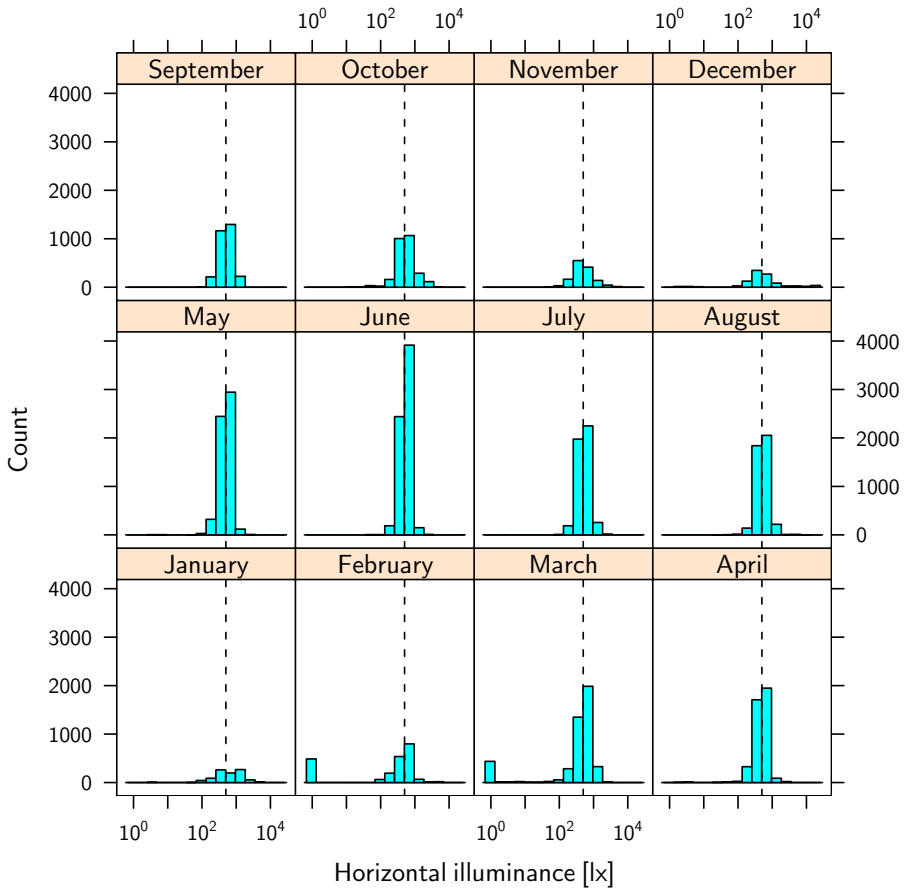


Figure 4.17: Illuminance distribution with daylight controller, per month. The dashed line on each panel corresponds to 500 lx.

5 Bayesian discomfort model

> While the systems were being installed a number of people who were going to work in the buildings found themselves having conversations with Breathe-o-Smart systems fitters which went something like this:
> 'But what if we want to have the windows open?'
> 'You won't want to have the windows open with the new Breathe-o-Smart.'
> 'Yes but supposing we just wanted to have them open for a little bit?'
> 'You won't want to have them open even for a little bit. The new Breathe-o-Smart system will see to that.'
> 'Hmmm.'
> 'Enjoy Breathe-o-Smart!'
> 'OK, so what if the Breathe-o-Smart breaks down or goes wrong or something?'
> 'Ah! One of the smartest features of the Breathe-o-Smart is that it cannot possibly go wrong. So. No worries on that score. Enjoy your breathing now, and have a nice day.'
>
> *(Douglas Adams, The Hitch Hiker's Guide to the Galaxy)*

In this chapter we will discuss the *Bayesian* aspect of this work. The central element of our daylighting control algorithm is an estimation of the user's visual discomfort from a statistical study of that user's past behaviour. Bayes's theorem is applied to estimate a *Bayesian Visual Discomfort Probability* as a function of the illuminance distribution in the office room, and this discomfort estimator is the main tool with which the controller maintains a visually comfortable environment.

We begin in section 5.1 with a review of bayesian statistics and in section 5.2 we discuss how they can be applied in our case. In section 5.3 we carry out such an estimation for all of LESO's building rooms on the basis of their historic recorded illuminance data. Section 5.4 will discuss our findings and includes the formula we will use to combine more than one variable in our discomfort estimation.

5.1 Bayesian inference

Bayesian inference is what we do when we infer that A must be true because we have observed B and that A and B usually happen together. For example, if we see a lion at a circus show we can infer that it must be tame, because all tame lions we have seen were part of a circus show, and we have never seen a wild lion in such a show.

There is strong evidence that the human brain functions according to bayesian inference. Knill and Pouget (2004) review the available evidence for the so-called "Bayesian coding hypothesis", a school of thought that holds that the brain represents sensory information probabilistically and makes inferences bayesian in nature. Series of experiments have successfully demonstrated that the brain carries a built-in prior probability curve for different kinds of events, which is updated as new evidence becomes available.

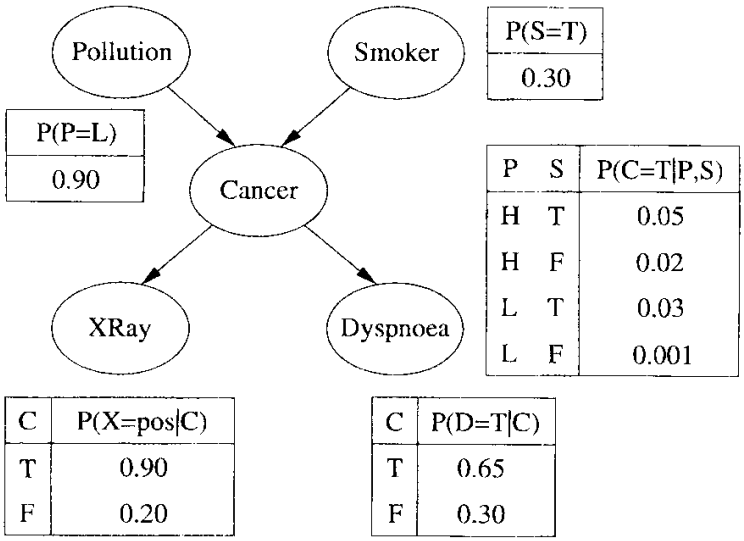


Figure 5.1: Bayesian network for lung cancer. P(·) stands for *Probability of* ·, C for *Cancer*, P for *Pollution*, S for *Smoker*, X for *X-Ray diagnosis*, D for *Dyspnoea symptom*, H for *High*, L for *Low*, T for *True*, F for *False* and pos for *Positive*. Adapted with permission from Korb and Nicholson (2003).

It was Reverend Thomas Bayes (1702–1761) who first discovered what is now known as Bayes's theorem: given two events, denoted by A and B, then the following holds:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)} \tag{5.1}$$

where $\Pr(A)$ stands for the probability of event A and $\Pr(A|B)$ stands for the conditional probability of A knowing that B has happened. $\Pr(B)$ can be expanded, yielding the same theorem in another form:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|\overline{A})\Pr(\overline{A})} \tag{5.2}$$

where $\Pr(\overline{A})$ stands for the probability of A not happening.

Bayes's theorem deals with only two events, but *Bayesian networks* link together an arbitrary number of events believed to exert a probabilistic influence on each other. Consider the following example, adapted from Korb and Nicholson (2003): a patient's chances of developing lung cancer are assumed to depend exclusively on whether they live in a polluted area, and on whether they smoke. Similarly, having cancer will determine the chances of an X-ray test to be positive and will also affect the chances of the patient developing a breathing condition known as dyspnoea. The probabilistic influences exerted among these events is shown in Figure 5.1.

Here the conditional probabilities are given explicitly, and successive applications of Bayes's theorem allow us to determine any other probability. For example, without knowing whether the patient exhibits dyspnoea and without the results of an X-ray test, the probability of any

patient having cancer (with symbols as defined in Figure 5.1) is:

$$\begin{aligned}\Pr(C=T) \;=\; & \Pr(C=T\big|P=H,S=T)\Pr(P=H)\Pr(S=T)\\ & +\Pr(C=T\big|P=H,S=F)\Pr(P=H)\Pr(S=F)\\ & +\Pr(C=T\big|P=L,S=T)\Pr(P=L)\Pr(S=T)\\ & +\Pr(C=T\big|P=L,S=F)\Pr(P=L)\Pr(S=F)\end{aligned} \tag{5.3}$$

which evaluates numerically to $\Pr(C=T)=0.012$.

But if the diagnostic of dyspnoea is positive, that probability increases and becomes, per the second form of Bayes's theorem:

$$\Pr(C=T\big|D=T)= \frac{\Pr(D=T\big|C=T)\Pr(C=T)}{\Pr(D=T\big|C=T)\Pr(C=T)+\Pr(D=T\big|C=F)\Pr(C=F)} \tag{5.4}$$

which evaluates to $\Pr(C=T\big|D=T)=0.025$, i.e. about twice the previous probability. (Note again that at no time did the patient suddenly develop cancer, but our *degree of belief* in cancer increased after learning about the positive dyspnoea diagnostic.) A very good tutorial on bayesian networks can be found in Heckerman (1995).

Bayesian inference has emerged in recent years as a particularly promising form of artificial intelligence and has gained a solid foothold in the medical world, where its use is facilitated by the existence of vast data archives, needed to derive probabilities such as the ones given in Figure 5.1.

Furthermore, since the publication of Paul Graham's seminal article (Graham, 2002), *bayesian classifiers*, i.e. bayesian networks which yield the probability of a particular node being true, have also become a major weapon in the fight against unsolicited junk email. The idea is to compute, with the first form of Bayes's theorem, the probability of an email being spam on the basis of the words the email contains. For example, if an email contains the word "Viagra", then the probability of the email being spam is given by

$$\Pr(\text{Spam}\big|\text{Viagra})=\frac{\Pr(\text{Viagra}\big|\text{Spam})\Pr(\text{Spam})}{\Pr(\text{Viagra})} \tag{5.5}$$

where Pr(Viagra) is the fraction of all the user's emails that contain the word "Viagra", Pr(Viagra|Spam) is the fraction of all the user's spam that contains that word, and Pr(Spam) is the fraction of all the user's emails that is spam. Note that these latter quantities require a preliminary training of the classifier by the user on a (preferably large) corpus of unfiltered email.

Statistics-based classifiers have proven remarkably successful at classifying email on the basis of the words the email contains, even with rather simple implementations of bayesian networks (Androutsopoulos et al., 2000). Bayesian classifiers find applications in many fields, including at least one fairly succesful attempt to play chess (Breyer, 2004). Sakkis (2004) and Robinson (2003) are excellent introductory texts on this topic.

The central claim of this chapter is that if (even naive) bayesian classifiers are so good at calculating probabilities of an email being spam, a classification hitherto believed to require human judgement, then they should also be able to calculate the probability for a certain visual environment of being comfortable or uncomfortable to its occupant. Such a classifier

should base its judgement on the physical variables it measures and classify the room as comfortable or not. In particular, this classifier will look for correlations between visual discomfort and the illuminance distribution in the zone.

The main idea of this work is to use bayesian methods to develop a building management system that optimizes a user's visual comfort. More specifically, we are going to derive a *user visual discomfort probability* given a set of physical variables. The task of the controller shall then be to keep that probability as low as possible.

5.2 User visual discomfort probability

Most commercial building controllers aim at keeping one or more physical quantities as close as possible to given setpoints. A heating controller will try to keep the room temperature as close as possible to a given value, usually adjusted by the user or chosen in advance by the designer or installer. An electric lighting controller will try to keep the indoor illuminance as close to a specified value as possible. There are also blinds controllers on the market that adjust the blinds' position according to simple rules based on the sun's position and/or the facade's vertical illuminance. These, however, take no account of individual users' preferences.

The development of better building control algorithms is still an active field of research. For instance, the building controller described by Guillemin (2003) keeps blinds' positions and electric lighting conditions close to values known to be preferred by the user, instead of explicitly controlling the resulting illuminance. It learns these values from the user's actions.

In our methodology we make the explicit assumption that there exists a (limited) set of physical, measurable variables that are sufficient to characterize the user's visual comfort. For the bayesian classifier to be efffective, these variables must be chosen to correlate well with visual comfort—they must change when the users act on the controls at their disposal. Such variables can include the horizontal workplane illuminance and the vertical (pupillar) illuminance at the user's eye level. The latter is, indeed, an input to the glare indices reviewed in section 2.4.3, which implies a correlation between visual discomfort caused by glare with the vertical eye-level illuminance, either explicitly (through a E_{v} term in the equation) or implicitly (through L_s terms).

The horizontal workplane illuminance is also likely to influence the user's comfort, or at least the user's visual performance. The *IESNA Lighting Handbook* (Rea, 2000) bases its recommendation of lighting levels in different settings principally on the horizontal workplane illuminance. The prescription of a satisfactory horizontal workplane illuminance goes back at least as far as Luckiesh and Moss (1937).

Additional variables can of course be added if found relevant. The identification of such variables that correlate with user discomfort is a research project of its own—the window luminance, in particular, might very well be one.

Of course, one must be careful not to mistake *correlation* for *causality*. Two variables might be strongly correlated without necessarily meaning that one has any influence whatsoever on the other. The canonical example is the prevalence of lighters in the homes of lung cancer sufferers. The correlation is very real and expected, but is due to a third, underlying common cause. Similarly, we must be careful when applying bayesian methods to only link together variables that can realistically be expected to be causally connected. For reasons explained in the text, we will assume such to be the case between the illuminance distribution and the visual discomfort.

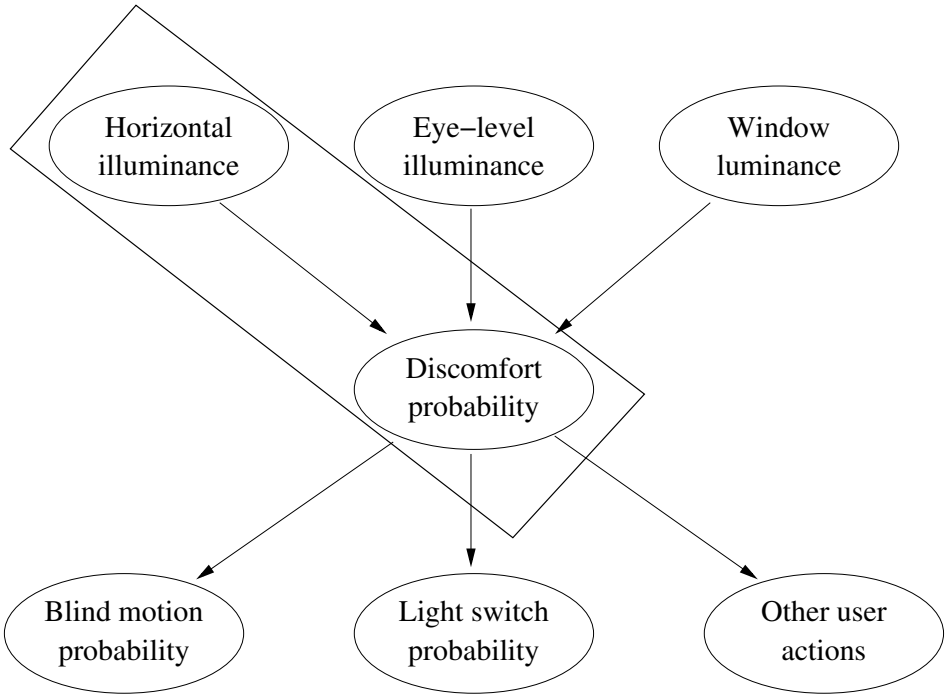


Figure 5.2: Example of a bayesian network for user visual discomfort. Only three physical variables are shown but there is no limit to how many can be taken into account. We will mainly focus on the relationship between horizontal illuminance and visual discomfort in this chapter, but will generalize our findings to more than one variable in section 5.4.3.

Such variables can then form a bayesian network that includes the probability that the user's visual environment is uncomfortable (see Figure 5.2). Also shown in this figure are events that could be derived from this discomfort probability, such as the probability that the user will act on the controls within the next five minutes. This topic would also deserve a research project by itself and is beyond the scope of this work.

There are two major differences between this network and the one shown in the lung cancer example. First, the nodes in the lung cancer example could only take discrete (boolean) values, whereas in our case, nodes such as "horizontal illuminance" take on continuous values. We should therefore rather speak of probability densities when dealing with continuous variables. For the sake of simplicity, however, we will continue to use our notation as if they were discrete.

Second, the probabilities associated with the "Cancer" node were explicitly given. They were derived from an extensive data-mining of existing medical archives. In our case, we must also do this statistical analysis in order to derive our user discomfort probability.

In this chapter we will first consider the case with only one physical variable, the horizontal workplane illuminance, and consider the case of multiple variables in section 5.4.3. We choose this variable instead of the eye-level illuminance (which, as argued above, correlates better with the sensation of glare) for two reasons. First, building construction codes usually prescribe horizontal workplane illuminances rather than eye-level illuminances, and it might be instructive to see if the prescribed values match the values preferred by the users. Second,

we will use historic data taken on the LESO building, in which only the horizontal workplane illuminance was recorded.

In an ideal world we would have had the possibility of presenting a user with a wide variety of combinations of blind positions, electric lighting intensities, and solar positions, and asking the user whether the situation was comfortable or not. From the results of such a survey, we would immediately obtain our desired probability of user discomfort as a function of horizontal illuminance.

Such a survey would, however, have to be re-done for every user (since we all have different preferences), for every room, and for every new variable under consideration. Furthermore, we do not have full control over all variables (e.g. sun position). Neither can we be expected to recreate in a laboratory setting the users' office rooms and ask them to evaluate hundreds of different situations.

Keeping in mind that we intend this method to be used by a controller in real office rooms, is there a data pool which can tell us the opinion of the user with certainty ? There is indeed one (probably the only one): *the set of situations immediately preceding and immediately following a user action provides us with a data pool of transitions from uncomfortable to (presumably) comfortable situations for that user.* Notice, however, that this assumption breaks down when several users have access to the same set of controls. Simultaneous visual comfort control for users placed at different positions with different orientations, perhaps with different sensitivities to glare, is notoriously difficult. It is therefore unclear whether our method is applicable to anything else than small office rooms with a single occupant.

If we denote by C the event *user comfortable*[1], by E the horizontal workplane illuminance, by T=*True* and F=*False* the possible values for C, and by e a possible illuminance value that E can take, we see that an application of the second form of Bayes's theorem (Equation 5.2) yields:

$$\Pr(C = F \big| E = e) = \frac{\Pr(E = e \big| C = F)\Pr(C = F)}{\Pr(E = e \big| C = F)\Pr(C = F) + \Pr(E = e \big| C = T)\Pr(C = T)} \tag{5.6}$$

Except for $\Pr(C = T)$ and its complement $\Pr(C = F)$, all the right-hand terms in this expression will be known from our data mining. For example, $\Pr(E = e \big| C = T)$, the illuminance distribution after user action, can be derived by histogramming the workplane illuminances resulting from the user action. There are, however, better approaches that will be described in section 5.3.

The $\Pr(C = F)$ term, also known to bayesians as the prior, has been the cause of much controversy in the statistical community. A couple of years ago the dust settled and the consensus seems now to be that in the absence of any prior information it is safe in most cases to set $\Pr(C = F) = \Pr(C = T) = 0.5$ (a justification for this will be given in section 5.4.2).

[1] To be absolutely rigorous, $\Pr(C = F)$ stands for the probability that, presented with a given visual situation, and explicitely asked whether that situation is judged uncomfortable, a user would answer affirmatively. We make in this paper the assumption that this interpretation is equivalent to saying that the situation is visually uncomfortable. Counter-intuitively, it is *not* the probability that the user is about to adjust the visual environment.

The preceding equation then simplifies to:

$$\Pr(C = F \big| E = e) = \frac{\Pr(E = e \big| C = F)}{\Pr(E = e \big| C = F) + \Pr(E = e \big| C = T)} \tag{5.7}$$

We shall now apply this method on data recorded on the LESO experimental building. The data used for this preliminary study is described in section 3.1.3. We have used the data recorded from mid-November 2002 to mid-January 2005.

5.3 Discomfort estimation on monitored data

5.3.1 User actions

A *user action* is defined as any action performed by the user on the blinds (raising or lowering) or on the electric lighting (increasing or decreasing intensity). Combinations of user actions spaced apart not more than one minute in time (such as the user fine-tuning the electric lighting level) are considered as part of the same user action. Actions performed less than two minutes before the user has left the office (such as switching the lights off for the day) are excluded.

A total of 7273 such user actions have been recorded. We have reconstructed the workplane illuminance before and after each user action. The illuminance measured by the sensors is discretized in steps of about 15 lx, so a random jitter uniformly drawn between −8 and +8 lx was added to the data. If the result became negative, its absolute value was taken.

5.3.2 Density estimation

We need first to determine $\Pr(E = e \big| C = T)$ and $\Pr(E = e \big| C = F)$ for each office. If E were a discrete variable we would simply count the number of times it realized each value and divide by the total number of events. But E is a continuous variable so it is strickly speaking a probability density we must estimate.

John and Langley (1995) discuss the value of different methods of density estimation in bayesian classifiers. The simplest density estimator is a classic histogram but the choice of bin width can influence the resulting density estimate. Other density estimators assume the data to be distributed according to some predefined models, often a gaussian.

In this work, we have no reason to impose any predefined model for either distribution. Some authors, for instance Fischer (1970) and references therein, have let users in a controlled laboratory setting adjust their horizontal workplane illuminance and have found the logarithm of this illuminance to be normally distributed. It might make sense to use this model at least for $\Pr(E = e \big| C = T)$, but for consistency and symmetry we will use the same approach for both distributions.

We will use a non-parametric density estimator, i.e. one that makes no use of any distribution model. The classical non-parametric density estimator is based on some kernel density estimates (Bowman and Azzalini, 1997), but even this estimator makes some assumptions on the underlying distribution. Following a recommendation from Sardy (2005) we use the "taut-string" non-parametric density estimator described by Davies and Kovac (2001), Davies and Kovac (2004) and Kovac (2007). It works as follows.

Let $x_1, x_2, \ldots, x_N$ be N observations whose density we want to estimate. Define $y(x)$ as the fraction of observations smaller than x, e.g. $y(-\infty) = 0$ and $y(\infty) = 1$.

$y(x)$ is a piecewise constant function. It increases by $1/N$ for each x_i. Its rate of increase is therefore large when the density of x_i is high. If we could smoothen that function, then its derivative would approximate that density.

To smoothen it, consider the functions $y_\mathrm{h}(x) = y(x) + C$ and $y_\mathrm{l}(x) = y(x) - C$. These two functions enclose $y(x)$. Now imagine a string being drawn to tightness between these two functions. For a large C, that string will be a horizontal line. The smaller C is, the closer to $y(x)$ that string will be. Somewhere in between, the string becomes an approximation to $y(x)$ whose derivative is our density estimate.

The details of this algorithm (in particular, its convergence criterion) are beyond the scope of this work. Figure 5.3 illustrates the algorithm's principle on an example "claw-shaped" distribution. Figure 5.4 shows the final density estimate on that distribution.

Errors on density estimates will be evaluated as follows. The probability density of a distribution at x, noted $p(x)$, is proportional to the number of elements n_x that would be drawn between x and $x + \delta x$, where δx is small enough, after N draws. We have:

$$p(x) = \frac{n_x}{\delta x N} \tag{5.8}$$

If δx and N constant, and assuming n_x follows a Poisson distribution, then the relative error $\Delta p(x)/p(x)$ on the probability estimate will be:

$$\frac{\Delta p(x)}{p(x)} = \frac{\sqrt{n_x}}{p(x)\delta x N} = \frac{1}{\sqrt{p(x)\delta x N}} \tag{5.9}$$

Intuitively, this means that the relative errors on the density estimate will be smaller when the density itself is high or when the total number of observations is large. δx is the bin width, that depends on the density estimation method being used. For a histogram, this bin width is constant. For the taut-string algorithm, we make the simplifying assumption that it too is constant, equal to an illuminance difference which does not significantly affect visual comfort. We arbitrarily set $\delta x = 100\,\mathrm{lx}$.

The relative error on a probability Pr calculated with Bayes's theorem, which is the ratio between a probability density p_1 and a sum of two probability densities p_1 and p_2, is then:

$$\frac{\Delta \Pr}{\Pr} = \Delta\left(\frac{p_1}{p_1 + p_2}\right) \Big/ \frac{p_1}{p_1 + p_2} = \sqrt{\left(\frac{\Delta p_1}{p_1}\right)^2 + \left(\frac{\Delta(p_1 + p_2)}{p_1 + p_2}\right)^2} = \frac{1}{\sqrt{\delta x N}} \sqrt{\frac{1}{p_1} + \frac{1}{p_1 + p_2}} \tag{5.10}$$

5.3.3 Single office room

> It is better to give a complete, baffling description than an incomplete, straightforward one.
>
> *(Donald E. Knuth)*

We will first carry out all the analysis on one single LESO office room, and then generalize to the other rooms. Office room 104 has had 983 user actions over the data acquisition period and is the office with the most user actions.

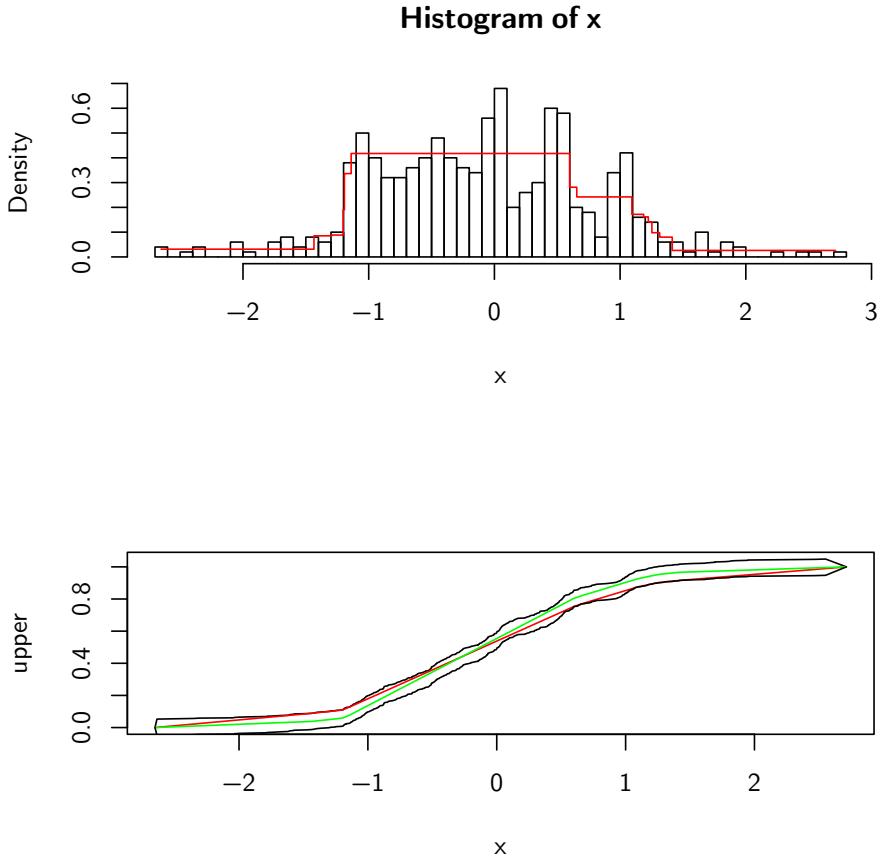


Figure 5.3: Intermediate step in the taut-string algorithm. This plot is produced by running the R taut-string implementation in verbose mode. The claw-shaped parent distribution is a sum of five narrow normal distributions added to a wider one. The histogram of the claw distribution, and the current density estimate, are shown in the top plot. The lower plot shows the y_{h} and y_{l} functions are they are being brought together. The red line is the current string drawn tightly from one extremity to the other. As they get even closer, finer and finer details of the density will be revealed, such as the five fingers of the claw-shaped distribution.

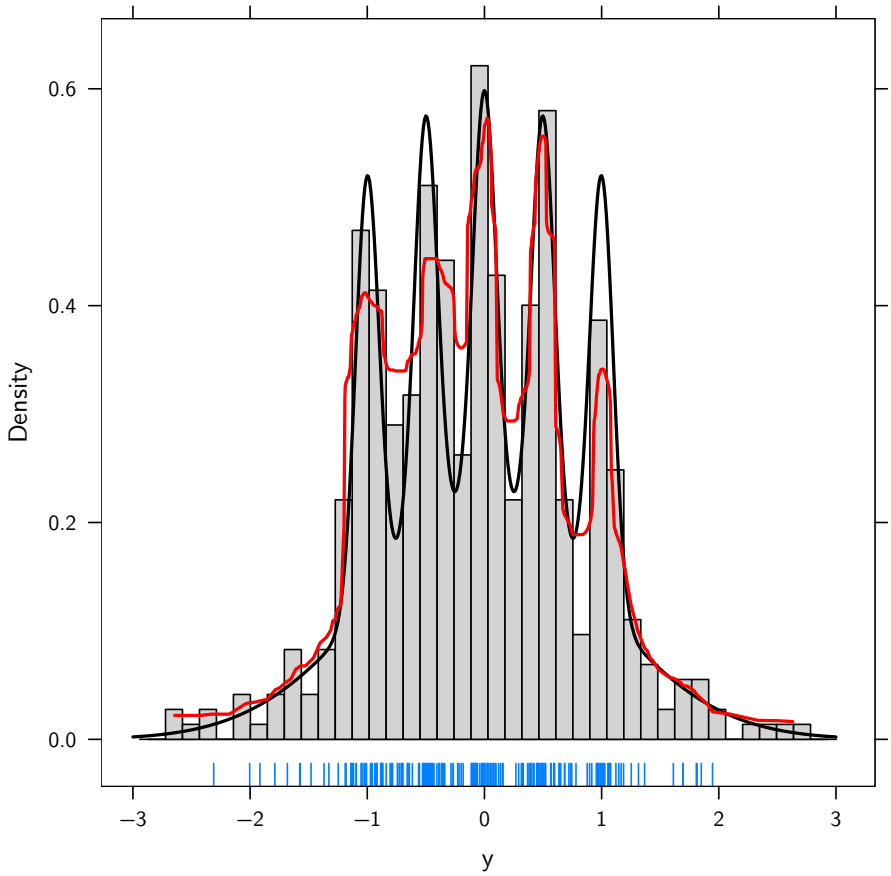


Figure 5.4: Example use of the taut-string non-parametric density estimator. The sample distribution is the same one as in Figure 5.3. The distribution estimated by the taut-string algorithm is shown in red. 200 data elements drawn from the sample distribution are shown as tick marks. A classical histogram is shown for comparison.

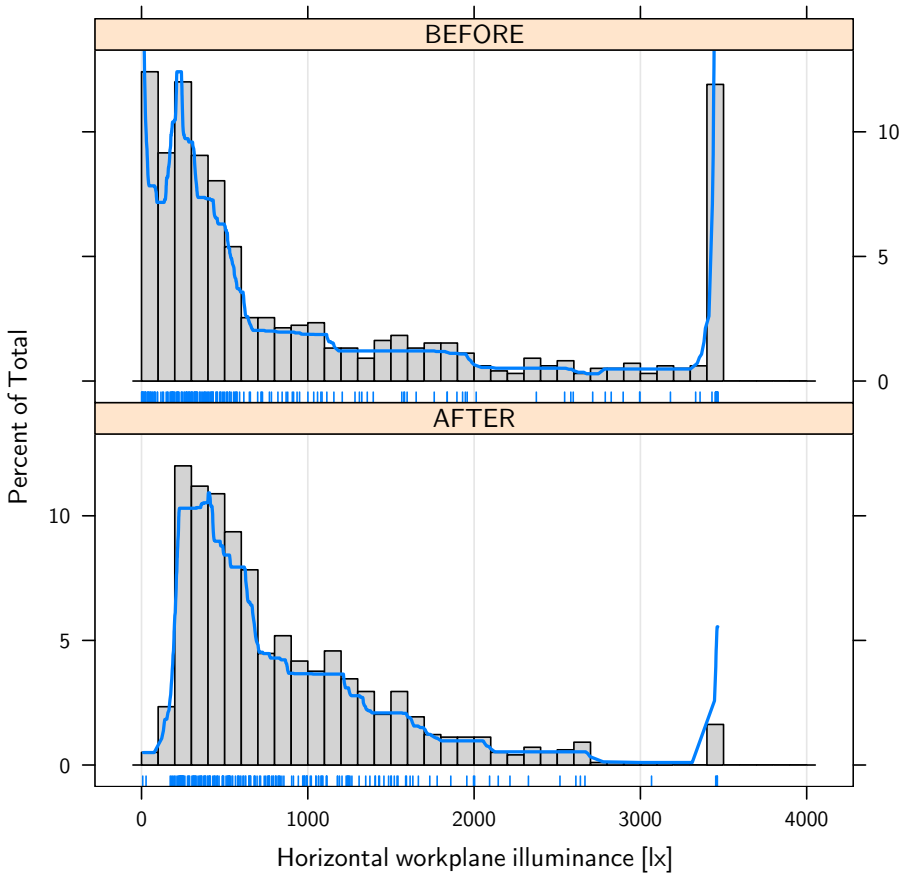


Figure 5.5: Illuminance distribution, before and after user action, in office room 104. User actions immediately followed by user exit are excluded. 200 randomly chosen illuminance values are shown on each panel as tick marks.

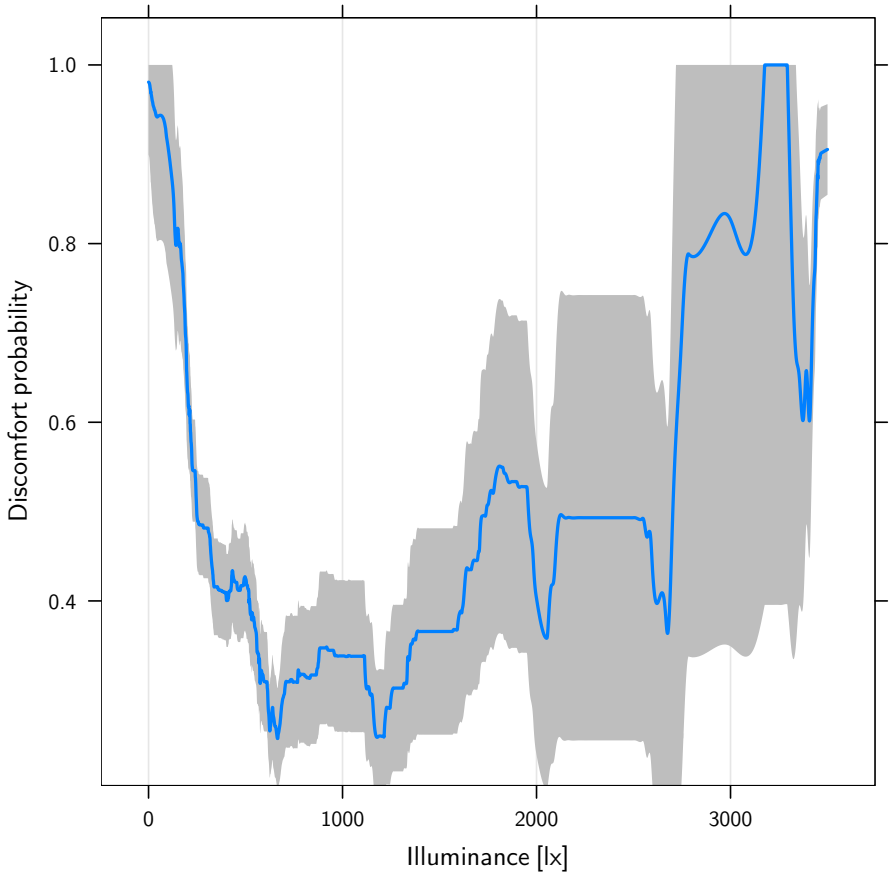


Figure 5.6: Discomfort probability as function of horizontal workplane illuminance, office room 104, with standard error.

In Figure 5.5 we show the density estimates together with histograms of the illuminances recorded immediately before and immediately after user actions in this office room. From these density estimates we compute, per Equation (5.7), the discomfort probability as a function of horizontal workplane illuminance, given in Figure 5.6.

The errors on this discomfort probability curve are small for illuminances below 1000 lx but grow larger for higher illuminances. This is caused by the smaller number of user actions for these higher illuminance values. Nevertheless, the general trend of this curve remains discernable. The discomfort probability is very high for low illuminance values. It reaches a minimum between 800–1200 lx (the zone of optimal comfort for this occupant), and increases again but at a slower rate until 3000 lx. It is difficult to interpret the behaviour of the curve beyond this illuminance because of the small number of user actions at these lighting levels. Notice also the sharp increase of the density estimates, caused by a saturation at 3500 lx of our ceiling-mounted luminance sensor from which our data comes.

Keeping in mind that we try to develop a discomfort estimator to be used in a daylight controller, and that this controller will most likely be commissioned without any data pre-recorded, how quickly will the discomfort estimate converge? How many user actions are

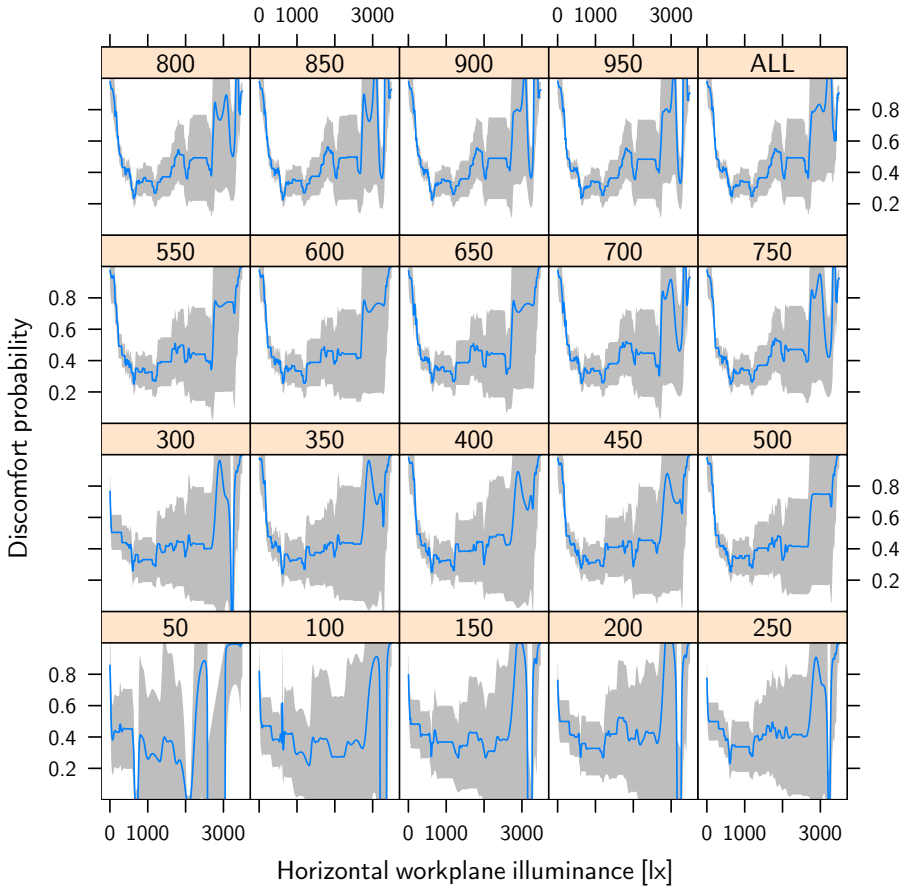


Figure 5.7: Evolution of discomfort probability estimate for office room 104. The number of user actions used for the estimate is shown in each panel.

necessary to obtain a discomfort probability curve precise enough to be used by the controller?

To answer this question, we select 50 user actions at random from our data and observe how the algorithm estimates the probability densities and the resulting user discomfort probability. Then we add a further randomly chosen 50 user actions from the remaining ones, recompute, then another 50 and so on until all user actions have been chosen. The resulting probability estimates are shown in Figure 5.7.

The probability estimate is always troubled by the lack of user actions at higher illuminances, but the presence of a global minimum around 1000 lx is already discernible after 150 events. Assuming an average of four user events per day, five days per week, this leads to an optimal user adaptation based on a single variable after 7–8 weeks.

5.3.4 Remaining office rooms

We now turn to the remaining office rooms of the LESO building. We show in Figure 5.8 and 5.9 the estimated densities of the workplane illuminances before and after the user action for all occupied offices of the LESO building. As before, the data points are represented beneath

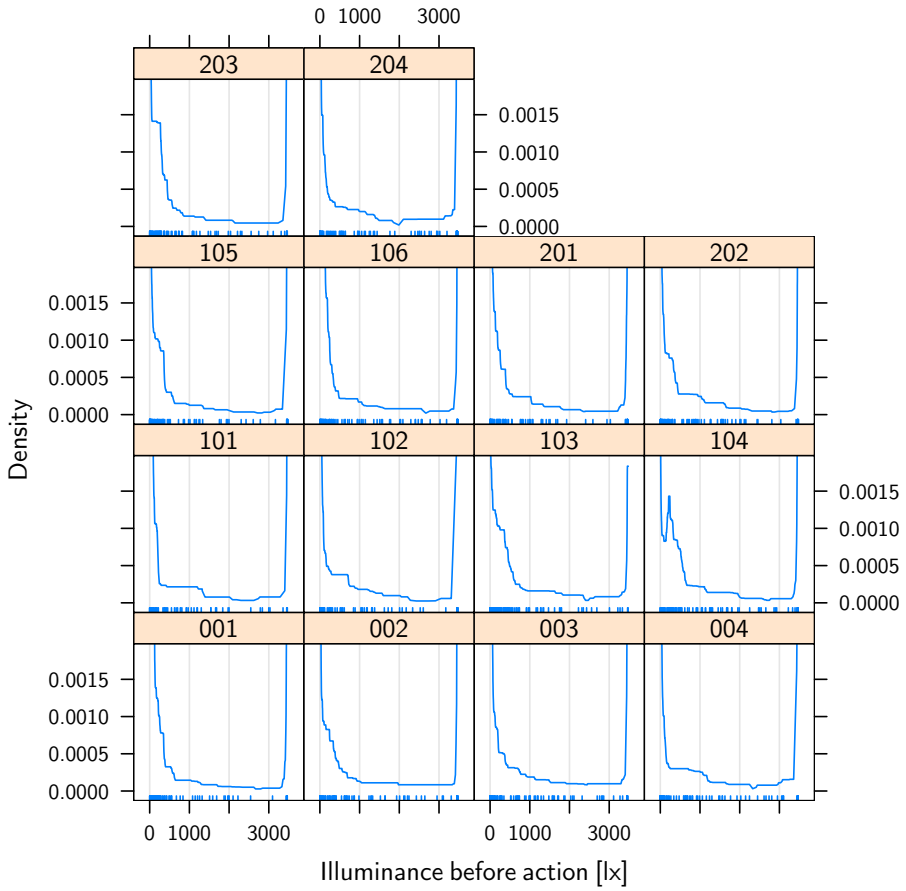


Figure 5.8: Density estimate of illuminance levels before user action, per office room. 100 random measurements are shown in each panels as tick marks. The numbers on each panel are the office rooms' identifiers.

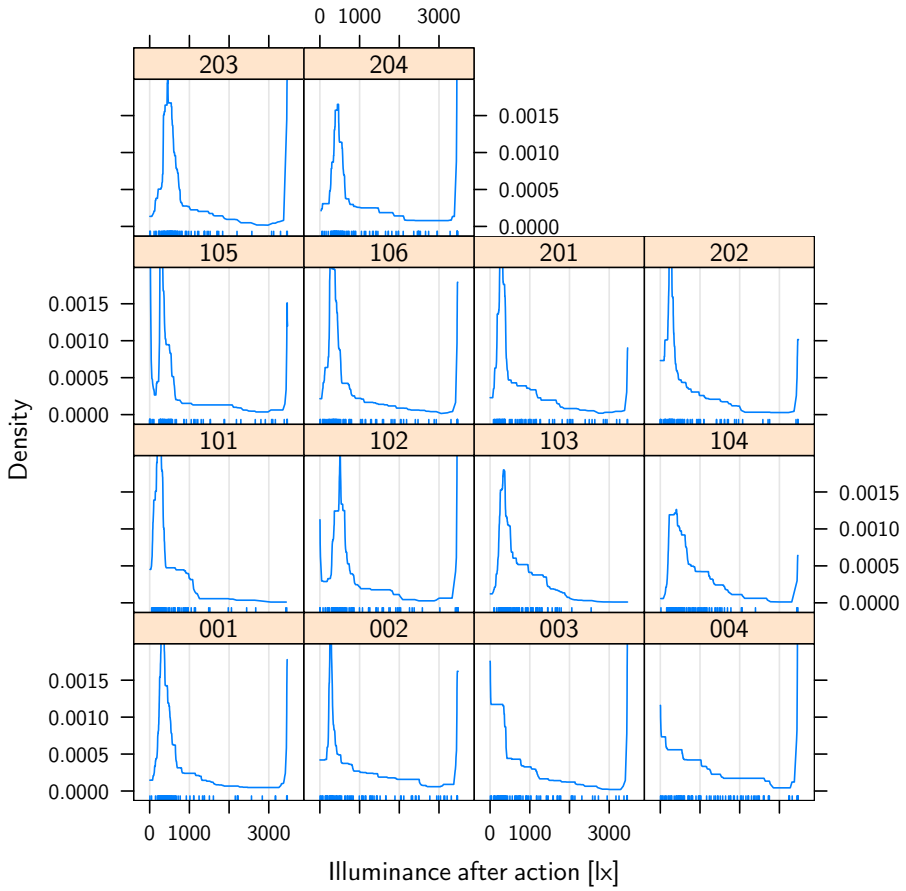


Figure 5.9: Density estimate of illuminance levels after user action, per office room. 100 random measurements are shown in each panels as tick marks. The numbers on each panel are the office rooms' identifiers.

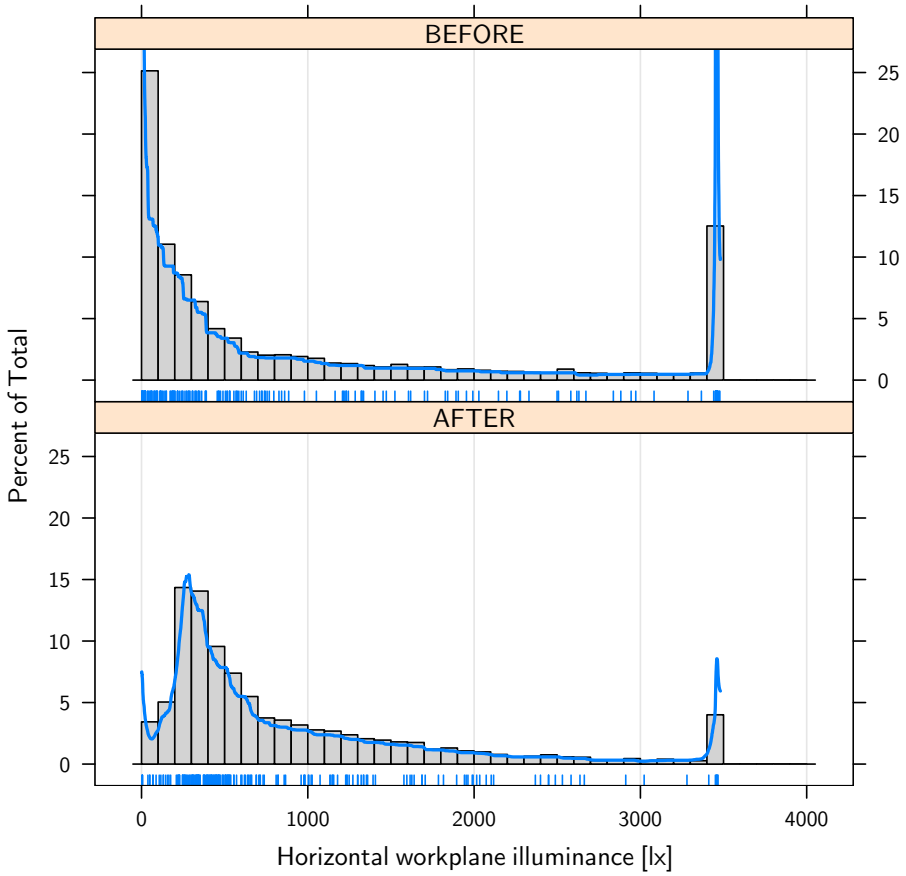


Figure 5.10: Density estimate of illuminance levels before and after user action, pooled data excluding offices 003 and 004.

each density curve as small ticks.

The users are remarkably consistent in that the illuminances most often seen to trigger a user action are either below 200 lx, or higher than 3000 lx. In other words, only very dark or very bright lighting situations prompt user actions. The peak seen on almost all plots at 3500 lx is due to the saturation of the illuminance sensor. However, this should not mean that most illuminance values between 200 lx and 3000 lx were optimally comfortable—users do not necessarily continuously adjust their blinds or their electrical lighting. They tolerate some minor discomfort that can depend on the design and placement of the controls at their disposal.

Similarly, the distribution of illuminances resulting from user actions tend to cluster around a value of about 400–500 lx. Again, the users are consistent among each other, with the possible exception of office rooms 003 and 004[2].

The consistency among the users allows us to lump together all the data, excluding office

[2]Office room 003 is the only office room at LESO with a complete anidolic mirror system (c.f. section 3.1). Office room 004 had an indirect lighting luminaire for some years right below the ceiling-mounted luminance sensor. These elements explain why the data from office rooms 003 and 004 show these differences.

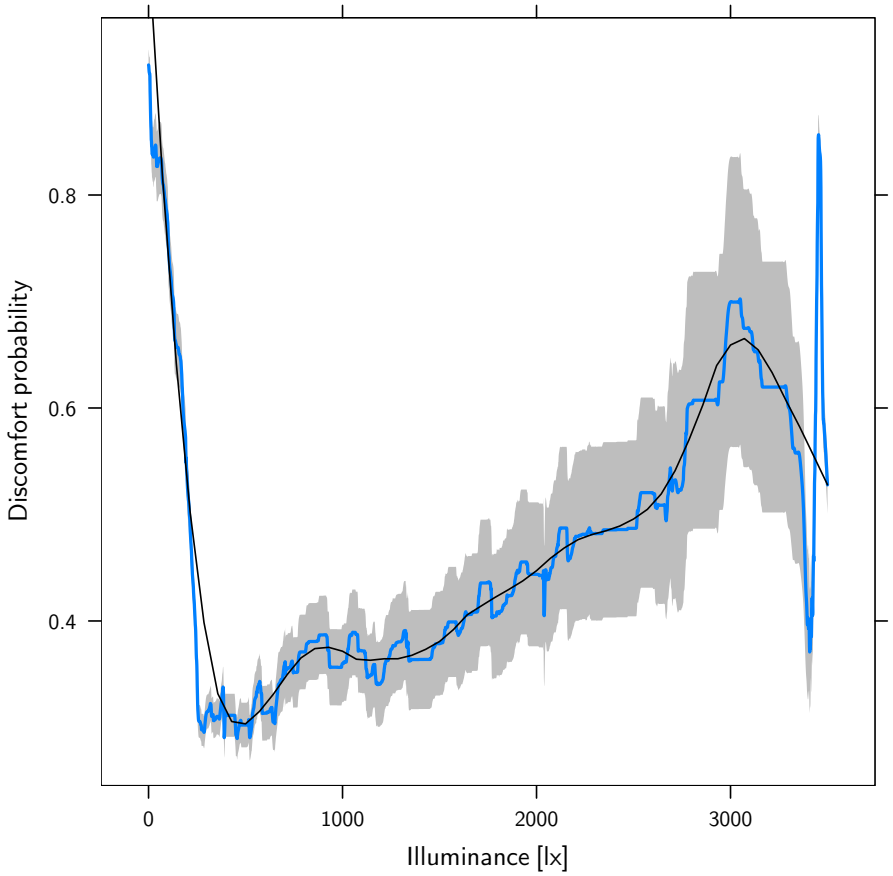


Figure 5.11: User discomfort probability as a function of horizontal workplane illuminance with standard error, pooled data excluding office rooms 003 and 004. The thin line is a approximation by the loess method (Cleveland et al., 1992).

rooms 003 and 004. The results are shown in Figure 5.10. The two curves shown on this figure correspond to our $\Pr(E = e|C = T)$ and $\Pr(E = e|C = F)$ respectively. From these two curves, we may now apply Bayes's theorem (Equation 5.6) and derive $\Pr(C = F|E = e)$, the probability of user discomfort as a function of workplane illuminance. That function, together with a smoothed approximation, is shown in Figure 5.11. The same curve but with different choices for the prior is given in section 5.4.2.

5.4 Discussion of results

5.4.1 Visual discomfort probability function

Figure 5.11 features a global minimum at about 500 lx. The user discomfort probability for lower workplane illuminances rises sharply, and would probably reach 1 were it not for inevitable measurements errors. We interpret this as meaning that 500 lx is the optimally comfortable workplane illuminance in LESO office rooms, which is consistent with the CIE

recommendations (CIE, 1986).

For illuminances larger than 500 lx, the curve rises gently until about 2500 lx. It is difficult to interpret the behaviour of the curve for higher illuminances because of the lack of user actions at these lighting conditions, but it is resonable to assume the discomfort probability reaches almost 1. Compare this with the simple calculation found in Michel (1999), who finds that the maximum horizontal workplane illuminance for reading/writing tasks should not exceed 4000 lx.

We interpret this to mean that users react strongly against too low illuminance. Once the available lighting is sufficient, the user is relatively indifferent to the workplace illuminance as long as no glare occurs, which is probably what happens for large workplane illuminances.

Note also that the minimum of this curve never drops below about 0.3. One would be tempted to interpret this as meaning that it is impossible to satisfy more than about 70% of the users. This interpretation is however incorrect for two reasons. First, we have calculated the probability that the user would judge the visual environment as uncomfortable *if prompted to do so*. This does not necessarily mean that the visual environment is so uncomfortable that the user will manually adjust the blinds' position. Second, remember that we compute this probability on the sole basis of the horizontal workplane illuminance. As was the case with the lung cancer example, a probability computed only on the basis of partial information does not necessarily reflect the "real" probability, in the classical sense.

A visual comfort controller that attempts to balance visual comfort with energy savings can use this probability function. It is indeed obvious from this curve that for our average user the preferred horizontal illuminance should be kept at 500 lx, and that higher illuminances can be tolerated (helping, for instance, the building use solar gains) until about 2500 lx, a limit that should never be exceeded.

5.4.2 Choice of prior

We have so far applied Equation 5.6 with the simplifying assumption that the prior $\Pr(C = F)$ is equal to 0.5. This assumption deserves some justification.

A choice of prior always reflects some information we have about a system before making observations. For example, let us consider the toss of a coin, which might or might not be loaded. Let us say we believe there is a 0.1 probability of the coin being loaded in such a way that is always lands on heads.

If after the toss the coin indeed lands on heads, then our prior belief that the coin was loaded should be reinforced. Indeed, applying Bayes's theorem, we obtain:

$$\Pr(\text{Load}\big|\text{Heads}) = \frac{\Pr(\text{Heads}\big|\text{Load})\Pr(\text{Load})}{\Pr(\text{Heads}\big|\text{Load})\Pr(\text{Load}) + \Pr(\text{Heads}\big|\text{NoLoad})\Pr(\text{NoLoad})} \tag{5.11}$$

which now evaluates numerically to 0.18 instead of 0.1.

In the case of user comfort, we should ask ourselves what is the prior probability of the user being uncomfortable in the total absence of any information. But let us put this question another way. Suppose you are given a coin, which might or might not be loaded, and asked to evaluate the probability of it landing on heads on the next toss. In such a complete absence of information, symmetry dictates that you should answer 0.5[3].

[3]The answer 0.5 is the right answer even if you are *explicitly* told that the coin is loaded one way or the other,

Similarly, a user placed in a completely unknown environment might or might not be uncomfortable, and our inability to favour one answer or the other compels us to choose 0.5 as a prior. This is the choice we have made in this work.

It might be argued, however, that a building built according to sound principles will usually ensure some degree of comfort (both visual and thermal) to its users. In these cases a choice of a prior different from 0.5 might be justified. How do different priors affect our posterior discomfort probability function?

Recall the expression for the discomfort probability, from Equation (5.6):

$$\Pr(C = F \big| E = e) = \frac{\Pr(E = e \big| C = F)\Pr(C = F)}{\Pr(E = e \big| C = F)\Pr(C = F) + \Pr(E = e \big| C = T)\Pr(C = T)} \tag{5.12}$$

For simplicity, let:

$$\begin{aligned} P &= \Pr(C = F \big| E = e) \\ C &= \Pr(C = F) = 1 - \Pr(C = T) \\ A &= \Pr(E = e \big| C = F) \\ B &= \Pr(E = e \big| C = T) \\ K &= B/A \end{aligned}$$

We have then:

$$P = \frac{A \times C}{A \times C + B \times (1 - C)} \tag{5.13}$$

or equivalently:

$$P = \frac{1}{1 + K \times \frac{1-C}{C}} \tag{5.14}$$

Deriving with respect to the prior C yields:

$$\frac{\mathrm{d}P}{\mathrm{d}C} = \frac{K/C^2}{\left(1 + K \times \frac{1-C}{C}\right)^2} = K \times \left(\frac{P}{C}\right)^2 \tag{5.15}$$

This shows that the derivative of the discomfort probability with respect to the prior is always positive, which we intuitively expect: if the prior probability increases, so should the resulting discomfort probability. It shows also that the derivative is a monotonous function in the discomfort probability P: if $P(E = e_1) > P(E = e_2)$ for two illuminance values e_1 and $e2$, then $\mathrm{d}P(E = e_1)/\mathrm{d}C > \mathrm{d}P(E = e_2)/\mathrm{d}C$. The larger probability will increase faster than the smaller one when the prior increases. The ranking between two arbitrary probabilities will always be conserved. In particular, the position of local and global minima and maxima are independent of the choice of prior.

Figure 5.12 shows two discomfort probability functions derived in the same way as the one in Figure 5.11, but with a prior choice of 0.9 and 0.1 respectively. Comparing these curves with the one with prior 0.5 given in Figure 5.11, we see that a different choice of priors does not affect the shape of the probability curve but tends to "squash" it to higher or lower values.

but without specifying which.

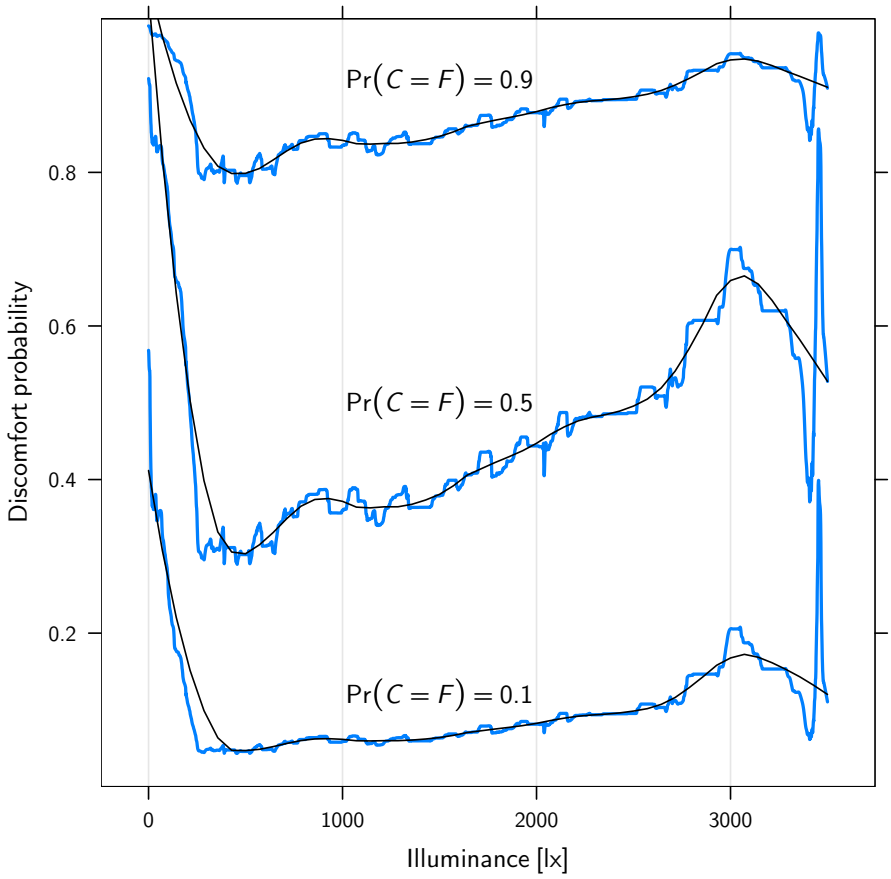


Figure 5.12: User discomfort probability for three different prior choices. The thin lines are smoothed loess approximations.

In conclusion, a controller that only attempts to minimize this discomfort probability will behave identically no matter what the choice of prior is. It is only when attempting to balance this discomfort probability with other factors, as is done in this project, that a non-trivial choice of prior can lead to different behaviours. It is however very unclear to us how to justify such a non-trivial choice of prior.

5.4.3 Bayesian network with more than one variable

It would be tempting to conclude that we now know how to satisfy the user on the basis of the horizontal workplane illuminance alone. Indeed, this was the whole point in using bayesian statistics: to extract the maximum amount of information from the available data.

We must not forget, however, that other factors can influence user comfort than just the horizontal workplane illuminance. The user can experience direct sunlight without the horizontal illuminance being affected, for instance. Therefore it is important that we make use of as many other indicators of visual discomfort as possible. The risk of direct sunlight, for instance, will be greatly reduced if we derived the same probability curve for the vertical eye-level illuminance.

We have so far considered only one physical variable, the horizontal workplane illuminance. Indeed, it was the only variable readily available in the office rooms of the LESO building. But a strength of bayesian networks is their ability to combine information from more than one node, i.e. to refine the posterior probability as more information becomes available. If we measured them, could we use the illuminances recorded in other locations in the office room, especially if some of these locations were known to highly correlate with visual comfort?

As we discussed in section 3.1, our project has fitted two office rooms with venetian blinds instead of textile blinds, and has replaced the ceiling-mounted luminance sensor with a calibrated desktop sensor. We have also added two additional sensors in each room on the walls at eye-level in front of, and behind, the user. We thus have had two additional variables believed to be correlated with user comfort, the vertical illuminance at eye level and the vertical illuminance in front of the user (taken to be proportional to the average luminance in the user's field of view).

An abundant discussion of methods to combine pieces of evidence in a Bayesian classifier is available since Paul Graham (Graham, 2002) wrote his influential article that sparked the advent of bayesian junk email classifiers. Here we shall describe only the simplest of them.

Elkan (1997) reviews the concept of a naive bayesian classifier, i.e. one where all pieces of evidence are considered to be conditionally independent of each other. If E_i are k physical variables under consideration, let $p_{iT} = \Pr(E_i = e_i \big| C = T)$ and $p_{iF} = \Pr(E_i = e_i \big| C = F)$ be the probability densities for each physical variable under comfortable respectively uncomfortable situations.

Applying Bayes's theorem, we see that:

$$p = \Pr(C = F \big| E_1 = e_1, \ldots, E_k = e_k) = \left(\prod_{i=1}^{k} p_{iF} \right) \times \Pr(C = F)/z \tag{5.16}$$

and

$$q = \Pr(C = T \big| E_1 = e_1, \ldots, E_k = e_k) = \left(\prod_{i=1}^{k} p_{iT} \right) \times \Pr(C = T)/z \tag{5.17}$$

where $z = \Pr(E_1 = e_1, \ldots, E_k = e_k)$ is a constant. But $p = 1 - q$ so:

$$\frac{p}{1-p} = \frac{p}{q} = \frac{\left(\prod_{i=1}^{k} p_{iF}\right) \Pr(C = F)}{\left(\prod_{i=1}^{k} p_{iT}\right) \Pr(C = T)} \tag{5.18}$$

and thus:

$$p = \frac{1}{1 + \frac{\left(\prod_{i=1}^{k} p_{iT}\right) \Pr(C=T)}{\left(\prod_{i=1}^{k} p_{iF}\right) \Pr(C=F)}} \tag{5.19}$$

Or even simpler, if we assume the trivial prior $\Pr(C = F) = 0.5$:

$$p = \frac{1}{1 + \frac{\left(\prod_{i=1}^{k} p_{iT}\right)}{\left(\prod_{i=1}^{k} p_{iF}\right)}} \tag{5.20}$$

which is exactly what we are looking for, the probability after combining all available variables.

Out of curiosity we have applied this calculation to the example given in section 5.1 to calculate the probability of a patient having cancer when smoking and living in a polluted area. The calculation yields a result of 0.077, slightly different from the given value of 0.05. The discrepancy comes because the numbers as given in the example are not compatible with a conditional independence of the evidences.

This conditional independence is actually almost *always* violated in practice. Naive bayesian classifiers work nevertheless surprisingly well. Domingos and Pazzani (1996) gives a number of interesting arguments to explain why even a bayesian classifier contrived to completely violate that independence will still perform well. For that reason, it is reasonable to use the naive bayesian classifier to combine evidences coming from such obviously dependent variables as illuminances measured from different points in the room. Equation 5.20 is the expression that our controller has used.

The other LESO office rooms have never had more than one illuminance sensor, so this method has not been validated on historical LESO data. We will instead use the results from this project and see in section 7.2 the results of a bayesian estimation of visual discomfort with three variables.

5.5 Chapter summary

Based on recorded user actions, we have derived the user discomfort probability function for occupants of the LESO building on the basis of the horizontal workplane illuminance. Its minimum lies around 500 lx, rises sharply for lower illuminance values, rises more gently for higher values until it plateaus at about 2500 lx.

This probability curve was obtained on-site, in a real, occupied building, with no interference with the regular workflow of its occupants. That curve has adapted itself to the users and can be used by a user-adaptive building management system. In particular, the integration of the discomfort estimation algorithm we have described in this chapter with our daylighting controller will be described in chapter 6.

The advantage of the bayesian approach is that we know exactly what we are estimating. Other visual discomfort or glare indices yield "scores", but it is difficult to interpret such scores.

No one can know from first principles what a Unified Glare Rating of 25 means without looking it up. A probability, on the other hand, has a meaning devoid of any ambiguity.

The complete dependency on the user's behaviour is, in this approach, not a weakness. It is indeed known, for instance from Foster and Oreszczyn (2001) and the references therein, that the occupant's use of venetian blinds is neither rational nor energetically optimal. A control algorithms that rely too much on learning from the user's behaviour run the risk of learning bad control strategies. It is, however, also known (Reinhart, 2001) that most users behave in a conscious and consistent way. In other words, it makes sense for a controller to learn from the desired *effects* of the occupants' actions, not necessarily from the actions themselves. Users that close the blind in the morning because the sun is low on the horizon, and leave the electric light switched on the whole day will usually (albeit not always) accept a system that shuts the lights off and opens the blinds when the source of glare is gone, even though they would not have bothered to do so themselves.

In our approach we seek to understand what lighting conditions are acceptable by a given user. The controller then uses this understanding to reproduce these lighting conditions as faithfully as possible. It is therefore quite different from the approach of Guillemin (2003), whose control algorithm explicitly optimizes a set of rules that reproduce, within limits, the user's behaviour. This latter control algorithm, while minimizing the risk of user rejection, will be more vulnerable to an irrational user's behaviour.

This approach can be applied to any variable linked to the occupant's comfort. Wong et al. (2007) have proposed a very similar algorithm applied to the thermal comfort. Whenever a user expresses a "complaint" about the indoor temperature, the temperature is logged and used in a bayesian algorithm to determine the optimal temperature for all users.

The application of bayesian statistics on building physics problems is clearly an emergent field, and we expect it to play an increasingly important role in the future.

6 Controller implementation

> Choose two: (A) Fast (B) Efficient (C) Stable (D) Windows 98 (counts as two)
>
> *(Anonymous)*

We have built a blinds and electric lighting controller that integrates the tools described in the preceding chapters. This controller provides an optimal visual comfort to the user while minimizing the use of electric lighting and maximizing the use of free solar gains.

This controller has been implemented as a proof-of-concept in an office room of the LESO building, interfacing with the existing EIB building management system. The objective was to prove the algorithm adapts itself to the user and performs as expected. Energy consumptions by that office have consequently been continuously monitored.

We have also run this control system against different virtual office rooms. No user adaptation was possible in this case, and we have instead focused on studying the predicted energy savings.

This chapter describes the technical implementation of the controller. We begin by reviewing in section 6.1 the most important requirements the controller must fulfill. In section 6.2 we discuss the implementation of the most important elements of our controller, such as the integration of the discomfort probability estimator, the choice of an optimization algorithm, and so forth. In section 6.3 we discuss the building management system interface we have built into EIBSERVER and which is also implemented by our virtual SIMBAD model. This interface makes it possible to run our controller indifferently against a real or a virtual office room. Section 6.4 describes technical design and implementation decisions.

After reading this chapter, the reader should have enough background information to understand the controller's source code (c.f. Appendix C). Some details will inevitably be skipped for simplicity's sake. Whenever this chapter and the source code seem to disagree, the source code has, of course, precedence.

This chapter is very technical in nature. Sections 6.3 and 6.4, in particular, are intended for a Java developer who needs to understand the controller's inner workings. They document what the reader needs to know in order to maintain and extend the existing system. A solid grounding in object-oriented programming (OOP), and in the Unified Modeling Language (UML), is recommended for these two sections. Meyer (1997) is one of the best texts available on OOP, while Fowler (2003) is probably the best introduction to UML.

6.1 Controller requirements

> It must often be so, Sam, when things are in danger: some one has to give them up, lose them, so that others may keep them.
>
> *(J. R. R. Tolkien, The Return of the King)*

In this section we will give specific requirements for the control system. Most of these are generic and would apply to any controller, but some of them (e.g. user adaptation) are very specific to the current project.

This section is by no means exhaustive, but enumerates the most important user-visible requirements the controller should fulfill without loosing sight of the project's original vision.

6.1.1 Blinds and electric lighting control

The control system shall control the position and slat angles of the office's blinds, as well as the electric lighting. It shall be capable of handling one or more venetian blinds.

It shall not do so in a continuous manner, which would result in constant noise from the blinds' motors. New positions to the blinds' motors shall not be sent unless one of the following conditions is satisfied:

- the user enters the office after a prolonged absence, or
- the user leaves the office, or
- at 15 min intervals, or
- if there is an immediate risk of glare.

Under no circumstance will a new command be sent to either the blinds or the electric lighting if the last user override has happened less than one hour ago and the user is still present.

6.1.2 Visual comfort

When the user is present, the controlled actuators shall provide an illuminance distribution in the office room such that the estimated visual discomfort probability is kept at a minimum.

If several combinations of blinds' positions and/or slat angles result in the same estimated visual discomfort, the combination that maximizes the view to the outside shall be preferred.

6.1.3 User adaptation

Following the methods described in chapter 5, the controller shall keep a log of illuminance distributions monitored by the room's sensors immediately preceding and following user actions, unless that action occurs when the user exits the zone.

These measurements are to be analyzed by the bayesian discomfort estimation algorithm, which will provide a estimate of the visual discomfort probability adapted to the personal preferences of the user and specific to the controlled office room.

6.1.4 Energy savings

The system shall use the electric lighting only when the visual discomfort probability decrease it can provide justifies the energy cost of using it. When the illuminance provided by the electric lighting becomes less than 25% of the total illuminance, it shall be switched completely off. Indeed, from the discussion in sections 2.5.1 and 2.5.2, a 25% illuminance decrease is half the smallest decrement in illuminance that results in a significant subjective effect. We deem, therefore, such a decrease as having no harmful effect on the visual discomfort and as being always justified if the electric lighting is thereby switched off[1].

A bias towards the use of solar gains shall always be applied when the indoor temperature risks falling below 20 °C. Conversely, a bias against solar gains shall be applied when the indoor temperature risks rising above 26 °C. This bias shall be much stronger when the user is absent. When the user is present, this bias shall be damped in order not to conflict with the visual comfort requirement[2].

6.1.5 Building automation response time

The system shall operate in a timely manner. This author has personally had some experience with the popular X10 home automation system, but noticed a flaw that might limit the system's acceptance by certain users: there is a noticeable delay when the remote control is used to switch on the lights.

This delay is less than one second, but people have become so accustomed to instant response times that this delay might be considered as unacceptable by some users. There is evidence from the literature on human-computer interaction that response times longer than 100–200 ms are perceived as bad (Jacobson, 1990).

The EIB system, however, has fast response times and no user of the LESO building has complained about any perceived slow responsiveness. Besides, our controller does not need to respond particularly quickly to any events, except when the user enters the room. After an absence period, the controller must within a couple of seconds at most set the blinds and electric lighting in an appropriate way. This is particularly important in the evening, when a fast electric lighting switch-on time is expected.

6.1.6 Solar variability response time

The system shall be able to respond efficiently and promptly to abrupt changes in the available daylight.

Tomson and Tamm (2006) have studied the short-term variability of solar radiation in northern Europe. They have introduced a classification of skies as being either *stable* or *highly variable*, depending on the fluctuation of solar irradiance over time.

For highly variable skies, they found that the period between two large increments in solar irradiance ($\Delta G > 150\,\mathrm{W/m^2{\cdot}min}$) is distributed exponentially, with 80% of these periods being shorter than 10 min and 19% of them shorter than 1 min.

[1] Using fluorescent tubes at very low power is particularly wasteful of energy, since their ballasts draw a constant power that might be larger than the power effectively used for lighting.

[2] Early tests showed that the controller would open the blinds completely when then sun was low on the horizon and almost perpendicular to the facade. In this situation, the solar gains completely dwarf the user's visual discomfort. Through trial and error, we have assigned a much smaller weight to the control's heating aspects when the user is present.

During such sky conditions (such as a partially overcast sky) the controller shall therefore respond to changes in the sky conditions that might signify a risk of glare. A change that does not pose such a risk does not need to be reacted to.

6.2 Design notes

The controller is implemented as a program that spends most of its time listening to events on the building's sensors and actuators. At regular intervals, and in response to certain events, it explores its degrees of freedom, using its daylighting model to find the set of blinds' and electric lighting settings that satisfies its requirements.

In this section we discuss the most important elements of such an implementation. Sections 6.3 and 6.4 will discuss its more technical aspects.

6.2.1 Integration of the visual discomfort probability

We know from chapter 5 how to derive the discomfort probability from measurements of the horizontal workplane illuminance, possibly combined with other illuminance measurements. We know also from chapter 4 how to predict in advance what the resulting illuminance will be for any combination of blinds and electric lighting settings. How should we use this information in our controller? How can these tools be combined in an integrated daylighting and electric lighting controller?

If the task of the controller were to minimize the user's visual discomfort probability at all costs, it could on a regular time basis (e.g. every 5 minutes) use its internal illuminance model to explore its degrees of freedom, before finding the blinds and electric lighting settings that minimize the user discomfort probability.

However, in an extreme case, this might in wintertime lead to an office whose solar shadings are completely closed to protect the user from direct glare from a low altitude sun, taking no advantage of solar gains during a period of the year when they are most needed, and having the electric lighting turned on at full power.

The controller should instead balance the user discomfort with the cost of a given configuration in terms of energy. The easiest way to do this is to write a total cost function that the controller should attempt to minimize. This function should be a sum of at least two terms: one expressing the energy consumption, and the other expressing the user's visual discomfort, with a suitable weighting factor introduced to balance the two. The cost function used in this project was inspired by the one described in Ferguson (1990) and takes the following form:

$$U = W_1 P_{\text{el}} + W_2 P_{\text{th}} + W_3 \Pr(C = F) \tag{6.1}$$

where P_{el} is the power applied to the electric lighting, P_{th} is the back-up power necessary to keep the office room at a set temperature[3], and W_i are suitable weighting factors[4].

[3] This can be either power applied to heating elements, or power applied to active cooling systems. The LESO building is passively cooled, and in these situations this term could either be left untouched, just as if a cooling system existed, or it could be replaced with an estimation of user's thermal discomfort resulting from excessive solar gains.

[4] Several variations on this cost function have been considered for the experimental setup in the LESO building. In particular, a "View" term has been added to take into account a positive cost when the blinds occlude too much of the window, affecting the user's view to the outside, and in fulfilment of requirement 6.1.2.

6.2.2 User adaptation

The central element of the bayesian discomfort estimation described in chapter 5 is the monitoring of the illuminance at several points in the controlled room before and after every user action (except those made upon user exit).

Whenever a user action is detected by the controller, this illuminance distribution is logged to disk along with the time of the user action. The density estimate of each set of illuminances needs to be computed in order to apply Bayes's theorem. In chapter 5 we used the R implementation of the taut-string algorithm (Davies and Kovac, 2001).

We have not reimplemented this algorithm in Java. Instead, our controller makes calls to an executable R script that reads in a file with illuminance data. This script returns a density estimate for illuminances between 0–3500 lx. Density estimates beyond 3500 lx are set to zero.

When the density estimate is done for each set of illuminance data, the controller is notified and reloads the density estimate data it needs.

Thus, user adaptation happens immediately after each user action. The external calls to R, and the execution of the R script itself, take some seconds but are done only once for each user action. Furthermore, the controller does not need to respond to events while overriden, so this small delay is perfectly acceptable.

Naturally, in an embedded controller the taut-string algorithm, or any other density estimation algorithm, would have to be implemented in another language than R. R is a high-level interpreted language—just its compiled library is almost 2 Mb large.

6.2.3 Daylighting model optimization

The controller includes the same DAQ program as the prototype controller developed in section 4.6. On weekends, while the user is absent, the controller moves the blinds to n discrete positions and to m discrete slat angles, and records the resulting illuminance on each sensor for each setting.

Let us introduce some notation. We shall denote blinds $100 \times X\%$ open and with slats $100 \times Y\%$ open as being in a (X, Y) setting. Thus, a blind 40% open and with slats 50% open is in a $(0.4, 0.5)$ setting. $X = 0$ corresponds to having the blinds completely closed and $X = 1$ to having them fully retracted. $Y = 0$ corresponds to having the slats in a vertical (closed) position, $Y = 1$ to having them in a horizontal (open) position.

The settings for two venetian blinds (as on the LESO building) will be noted $[(X, Y), (Z, Q)]$. The first setting will be the lower blinds and the second will be the upper blinds. On the LESO building, the DAQ program sets the blinds' position to $0, 20, \ldots, 100\%$, i.e. $n = 6$ discrete positions. The slat angle is set to $0, 10, \ldots, 100\%$, i.e. $m = 11$ discrete steps.

Our implementation uses the fact that the daylighting flux is additive. It is not necessary for the DAQ program to explore all $n \times m \times n \times m$ possibilities. Instead, the DAQ program will keep one of the blinds completely closed while the other explores its $n \times m$ settings. When finished, the latter blind will close and the former one will begin exploring its degrees of freedom.

When asked to predict the resulting daylighting illuminance from two venetian blinds, the daylighting module will first model A, the illuminance that would result if the lower blinds were completely closed, then B, the illuminance that would result if the lower blinds were opened again but the upper blinds closed. The sum $A + B$ is the illuminance from both blinds but includes stray illuminance from the windows whose blinds are completely closed. This is

why we also model C, the illuminance that would result if both blinds were closed. The total illuminance will be $A + B - C$ (c.f. Figure 6.1).

If the blind, whose illuminance contribution we want to model, is in one of the discrete states recorded by the DAQ program, then the algorithm described in chapter 4 is followed. If it is not, the following procedure, illustrated in Figure 6.2, is followed.

First the four discrete blinds settings enclosing (X, Y) are found, denoted by $(\lfloor X \rfloor, \lfloor Y \rfloor)$ and so on. Their associated illuminances are computed through our algorithm. For example, if the blinds are in a $(0.55, 0.47)$ settings, then the illuminance contribution from the blinds in the $(0.4, 0.4)$, $(0.4, 0.5)$, $(0.6, 0.4)$ and $(0.6, 0.5)$ settings are computed. This operation is computationally expensive, but the illuminances for these discrete values are likely to be requested several times until the next irradiance sensor event. Therefore they are cached.

Then the intermediate illuminance values for $(X, \lfloor Y \rfloor)$ and $(X, \lceil Y \rceil)$ are linearly interpolated from the illuminances previously computed. These illuminances correspond to having the blinds' position kept in a X position but whose slat angles are rounded down, respectively up, to a discrete value. In our example, we would compute the illuminance contribution from a blind in the $(0.55, 0.4)$ and $(0.55, 0.5)$ settings.

Finally, the desired illuminance for the blinds' setting (X, Y) is linearly interpolated from these two latest values.

6.2.4 Solar vector computation

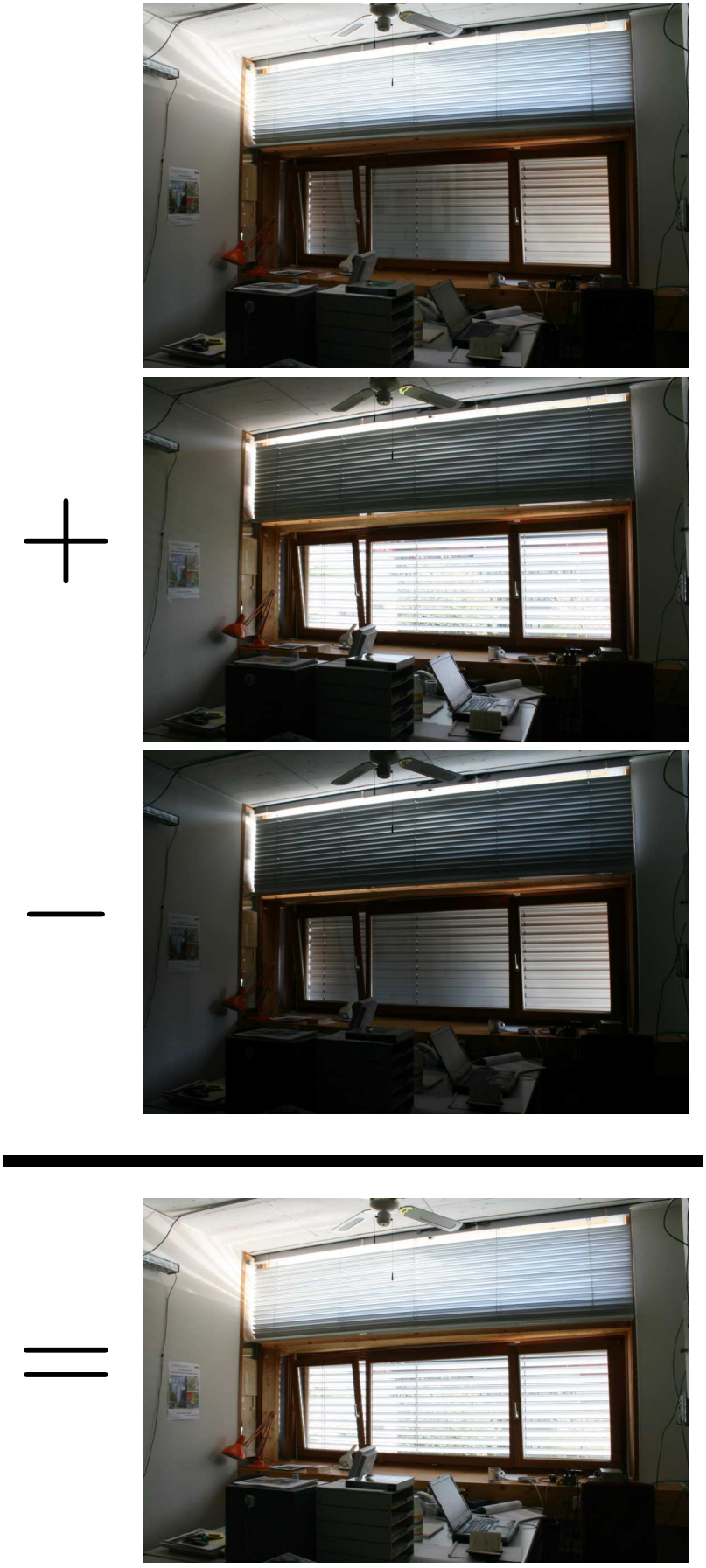
Our controller needs an accurate, computationally light algorithm for computing the solar vector, i.e. the sun's elevation and azimuth. Its accuracy needs to be such that negligible errors are made by the controller's daylighting model. Given the sensitivity of venetian blinds to the sun's height, especially when close to the cut-off angle, an accuracy better than the angular width of the solar disc (about $32''$) is required.

Blanco-Muriel et al. (2001) have reviewed seven algorithms for computing the solar vector with a sufficient accuracy to be used in high-concentration solar thermal systems. Of these seven, only four yield the true horizontal coordinates of the sun. They have evaluated the errors of these algorithms and found the *Astronomical Almanac*'s algorithm, described and implemented in Fortran by Michalsky (1988), to have the smallest average error on zenith angle ($-0.121''$), azimuth ($-0.042''$) and smallest average sun vector deviation ($0.207''$). Michalsky's algorithm is accurate between 1950–2050. Blanco-Muriel et al. further decribe their own proposed algorithm which is even more accurate. In this work, however, we have implemented Michalsky's algorithm, which is accurate enough for our application, simple (an R implementation needs about 55 lines of non-comment code) and computationally light[5]. Its R source code is given in Appendix C.3.

6.2.5 Optimization algorithm

The controller's main loop consists in minimizing a cost function with one parameter per degree of freedom to control. In our case, we have two venetian blinds with two degrees

[5]One line in Michalsky's Fortran implementation is wrong. The line that reads `eclong = mnlong + 1.915*(mnanom) + 0.20*sin(2.*mnanom)` should read instead `eclong = mnlong + 1.915*sin(mnanom) + 0.02*sin(2.*mnanom)`. The paper's main text is correct and follows the source of the original algorithm (Seidelmann, 1992).



119

Figure 6.1: Daylighting flux additivity.

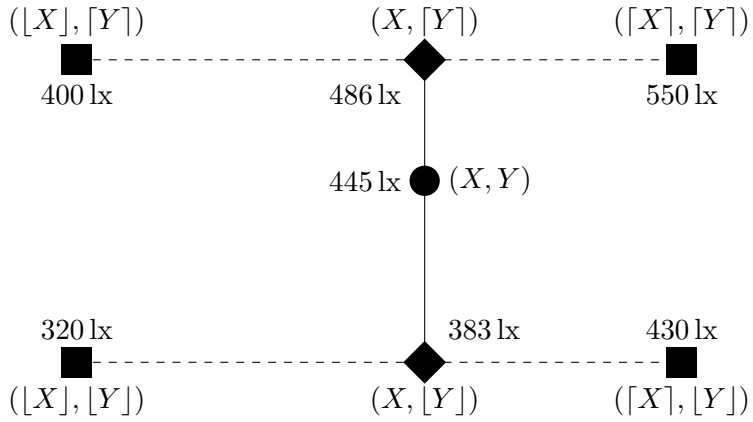


Figure 6.2: Daylighting model for non-discrete blinds settings. This diagram shows how the illuminance from a blinds' arbitrary (X, Y) setting is interpolated from the illuminances recorded in the discrete settings (solid squares) set by the DAQ program. Intermediate illuminances are first computed for the blinds settings shown as diamonds, from which the final illuminance is interpolated. In this diagram the ratio $(X - \lfloor X \rfloor)/(\lceil X \rceil - \lfloor X \rfloor) = 0.575$, and the ratio $(Y - \lfloor Y \rfloor)/(\lceil Y \rceil - \lfloor Y \rfloor) = 0.6$. Example illuminances are shown.

of freedom each, and the power applied to the electric lighting, yielding a function of five continuous parameters.

A brute-force search for a global minimum, although possible in principle, is unacceptable on the LESO offices. Even if we restricted ourselves to the 6 discrete values of the blind's positions and the 11 values for the slat angle, and 11 settings for the electrical lighting, we would have $6 \times 11 \times 6 \times 11 \times 11 = 47\,916$ points to evaluate. This is far too costly and would not yield a responsive controller.

We need an algorithm for minimizing a multivariate function, and we do not expect the latter to be particularly "misbehaved". Control settings close to one another should yield cost functions close to one another. One possible exception is when the slat angles are close to the cut-off angle, and a small change in slat angle can dramatically affect the indoor illuminances, and hence the user discomfort. But in general, the cost function will be continuously derivable, and have several local minima.

The literature has many multidimensional function optimization routines. Press et al. (2002), in their classic work, describe and implement four of the more traditional methods. Less traditional methods based on genetic algorithms or simulated annealing have recently gained much favor, especially in the building control algorithm community where the functions to optimize are often discontinuous.

Wetter and Wright (2004) have reviewed the performance of twelve different algorithms on optimization problems likely to be met by building simulation professionals. The best algorithm was a combination between a particle swarm optimization (PSO) and the Hookes-Jeeves algorithm.

In PSO algorithms, an initial population of candidate solutions (called *particles*) is randomly created. Each iteration of this algorithm models this population as a flock of birds or a school of fish:

> Each particle attempts to change its location to a point where it had a lower cost function value at previous iterations, which models cognitive behaviour, and in a direction where other particles had a lower cost function value, which models social behaviour.

The Hooke-Jeeves algorithm searches for a solution on a discrete mesh, looking for the point on that mesh that has the lowest cost function value. The algorithm refines the mesh as it progresses, making progressively larger steps in the direction that has reduced the cost in previous iterations.

A hybrid PSO and Hookes-Jeeves algorithm begins with a PSO on a mesh, then switches over to a Hookes-Jeeves algorithm using for the initial iterate the mesh point that attained the lowest cost function value.

Genetic algorithms are, however, also a good choice at the price of a slight decrease in accuracy. Two algorithms perform so badly that the authors explicitly recommend against using them: the Nelder-Mead and the discrete Armijo gradient algorithms.

Before we choose one algorithm or another, however, several points must be kept in mind. First, we want the optimization routine to be fast, yielding solutions in less than a couple of seconds as argued in section 6.1.5. Second, we do not expect discontinuities in the function to optimize. Third, the function has only five parameters, whereas cost functions encountered in building design problems may have up to 10–20 parameters (one of the two optimization problems used by Wetter and Wright had thirteen).

Fourth, and this is an essential point, we do not necessarily ask for a global minimum. When different combinations of blinds' positions and slat angles yield the same user comfort we want the algorithm to select an acceptable solution that is close to the current settings, in order to keep the distance moved by the blinds to a minimum. Too many large movements will inevitably annoy the user and wear the blinds' motors out. Therefore, even algorithms known to be vulnerable to local minima should not be excluded and might, for our purposes, be actually the right choice.

These reasons suggest that the warnings given by Wetter and Wright do not apply in our case, and we have chosen the Nelder-Mead algorithm for its speed and simplicity.

This algorithm starts out with $N + 1$ initial points, where N is the number of dimensions. These points define a geometrical figure called a *simplex*: in three dimensions, the simplex is a tetrahedron. The simplex then takes a series of steps, most of which consist in reflecting the highest point through the opposite face of the simplex to a lower point. Other rules are also tried, illustrated in Figure 6.3. The behaviour of the simplex mimicks that of some intelligent jelly that oozes down a $N+1$ dimensional surface: in two dimensions, the simplex is a triangle that oozes down along a three-dimensional surface, the altitude of each point being given by the cost function value at that point.

The main body of our controller's `Amoeba` class is our translation in Java of the C++ implementation given in Press et al. (2002, p. 413–417)[6].

6.2.6 Alternatives to a cost function

Several stakeholders in this project required different goals to be met by the control algorithm, and the cost function discussed in section 6.2.1 was proposed as a compromise. In this section

[6] We have however made minor modifications to that implementation so the function's variables would remain bounded.

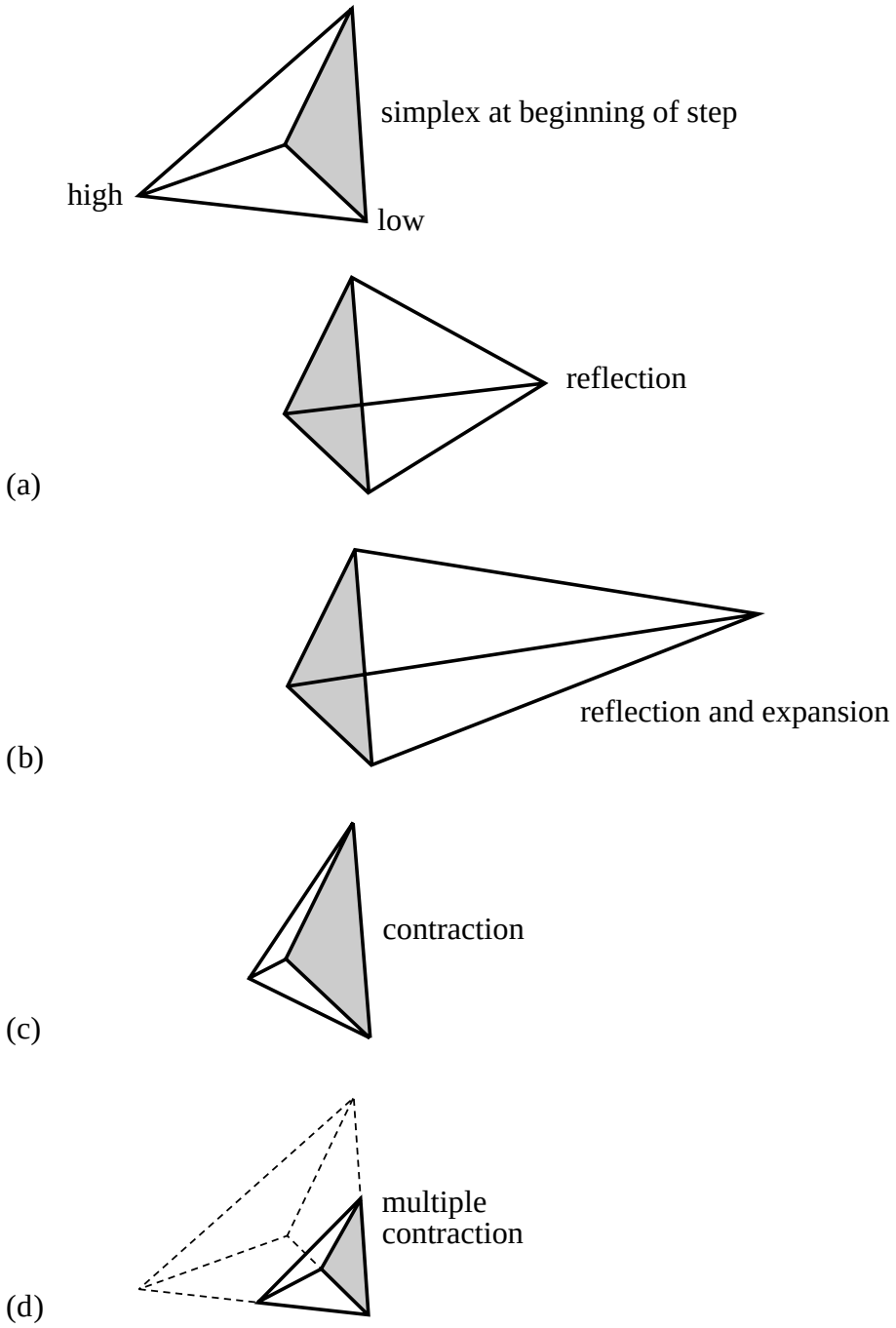


Figure 6.3: Rules followed by a simplex in its search for a minimum, from Press et al. (2002). The top figure shows the simplex at the beginning of each iteration step. From there, it can either a) reflect the high point through the opposite surface; b) reflect and stretch further away from the high point; c) contract away from the high point; and d) contract along all dimensions towards the low point.

we will discuss alternative possibilities that could be explored in future projects.

Our optimization problem is complicated by seeking to optimize two quantities at the same time: the user comfort and the energy savings. A cost function was chosen that balances the two terms, partly inspired by the function proposed by Ferguson (1990).

Other approaches are possible. Wright et al. (2002) discuss the problem of a so-called multi-criterion optimization problem and how they solve it through a multi-objective genetic algorithm. The advantage of their approach is that the optimization does not need any relative weights to assign to any criterion. The algorithm yields a family of optimal solutions, and the algorithm user selects then the solution that best accounts for the relative merits of the different criteria. In this case, however, a decision must still be made on the relative importance of the different criteria. The difference with our approach is that this decision is made after the optimization instead of before.

Perhaps the idea of a central cost function is not the best one either. The difficulty with any integrated controller is, by definition, to integrate many different and interrelated aspects of building control. But the control strategy for one category of building services can be strongly influenced by the controllability of other building services. For example, a building whose heating can be controlled will not use solar gains in the same way that our controller does: the latter must keep the indoor temperature at a given level at all times (c.f. section 6.1.4), whereas the former knows that heat is not necessary when the user is absent. And its seems very clumsy to this author to have to redefine a new cost function for each particular kind of controller.

Nevertheless, the central idea of this work was to provide building controllers with a tool that estimates the user's visual discomfort from a statistical analysis of the user's behaviour. The controller we have built is meant as a proof-of-concept implementation, not as a reference example of how this tool should be used.

6.3 Building bus interface

EIBSERVER was originally designed to communicate with a control algorithm implemented and running in MATLAB on another computer.

The de facto standard way for a Java program to communicate with other processes, possibly running on other machines, is the RMI (Remote Method Invocation) library. This library defines a `java.rmi.Remote` interface that a class must implement in order to make its methods transparently[7] available to the other process.

The architect of any RMI-based distributed application faces a dilemma. Imagine we are to design software for automated telling machines. These machines must be able to invoke remote procedures on a server running at the bank that manages the customer's accounts. But should there be one big remote server process representing the bank, or should there be many small servers processes each representing a customer's account? The pros and cons of either approach are discussed by Grosso (2001).

Early versions of EIBSERVER implemented the first solution, i.e. a huge `EibRmiServer` class that provided remote methods such as `getBlindPosition(String roomNumber, String blindName)`. Such methods served as middle-men between the controller process

[7] With some exceptions, inevitable in distributed computing. Besides obvious lag problems and loss of connectivity, all remote methods are declared to throw `java.rmi.RemoteException`s which must be checked by the caller.

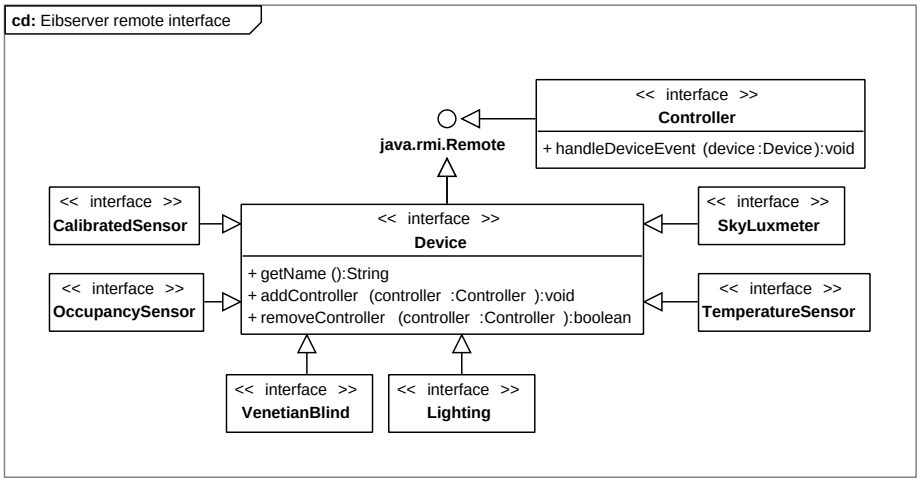


Figure 6.4: EIBSERVER remote interface class diagram.

and the EIBSERVER objects. The method above would, for instance, forward a request to the `Blind.getPosition()` method on the correct object.

This design had its flaws. Any modification in EIBSERVER, such as the addition of new classes representing new building services, required the developer to remember adding the corresponding methods to `EibRmiServer`. And there was no way a controller process could dynamically query EIBSERVER and ask for a list of available methods or devices.

When we took over the maintenance of EIBSERVER we tried to improve on this design. Instead of having one single object doing all the RMI communication, all device interfaces now extend a common `Device` interface, that itself extends the `java.rmi.Remote` interface, and each class's constructor exports the object upon instantiation. Thus all classes representing actuators or sensors implement the corresponding interface `Lighting`, `Temperature` and so on, each of which extends the `Device` interface, as shown in Figure 6.4. The public interface specific to each kind of device is shown in Figure 6.5. This public interface is accessible to any external process through RMI.

EIBSERVER also defines a `Controller` interface that a class must implement in order to subscribe to device events. The subscription is done by passing itself as argument to the `Device.addController()` method. Thereafter, any change of status will cause a call to `Controller.handleDeviceEvent()`, with the changed device being passed as argument.

To conclude this section, we give below a final sample script that will list all devices available from EIBSERVER's RMI server (only the first five items are shown, corresponding to a window opening sensor, two venetian blinds, and two lighting actuators). The device names are always given as `//lesopc7:1099/light_202` where `lesopc7` is the name of the machine on which EIBSERVER runs, 1099 is the port number used by RMI, and `light_202` is the RMI identifier of the object:

```
1 [lindelof@lesopriv3 eccobuild]$ jython
2 Jython 2.2a0 on java1.4.2_05 (JIT: null)
3 Type "copyright", "credits" or "license" for more information.
4 >>> from java.rmi import Naming
5 >>> list = Naming.list("//lesopc7")
```

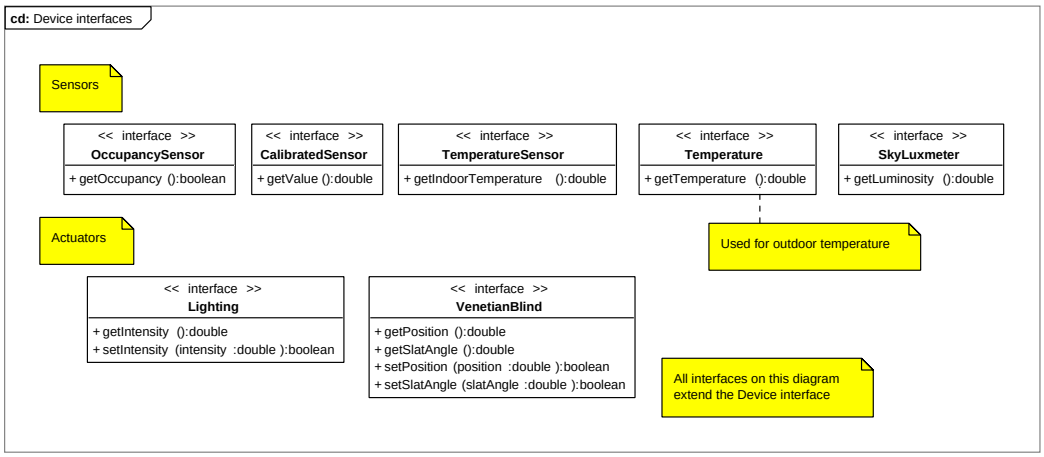


Figure 6.5: EIBSERVER sensors and actuators interfaces.

```
>>> for item in list:
...     print item
...
//lesopc7:1099/win-N_107
//lesopc7:1099/bldV-up_201
//lesopc7:1099/bldV-up_202
//lesopc7:1099/light_201
//lesopc7:1099/light_202
[lindelof@lesopriv3 eccobuild]$
```

6.4 Design decisions

> With design I can think very fast, but my thinking is full of little holes.
>
> *(Martin Fowler)*

6.4.1 Overall controller structure

The controller is implemented as a Java package, released as a Java executable `eccobuild.jar` jarfile. Figure 6.6 shows the deployment diagram of the controller on the LESO building. Figure 6.7 likewise shows the deployment diagram when running against the building simulation model. These figures show no difference as far as the control PC is concerned, and the same executable is run against the real and virtual building.

The class diagram shown in Figure 6.8 shows the main classes that make up the controller's implementation. The central class is `EccobuildController`, of which only one instance will exist per controlled room. That class implements the `Controller` interface, allowing it to register for event notification with the controlled `Devices`, as previously discussed in section 6.3.

As we will see shortly, our controller has state and needs to track that state through two enumeration classes, `PresenceStatus` and `ControllerStatus`. We will defer the discussion of these states until after we have described the main event loop.

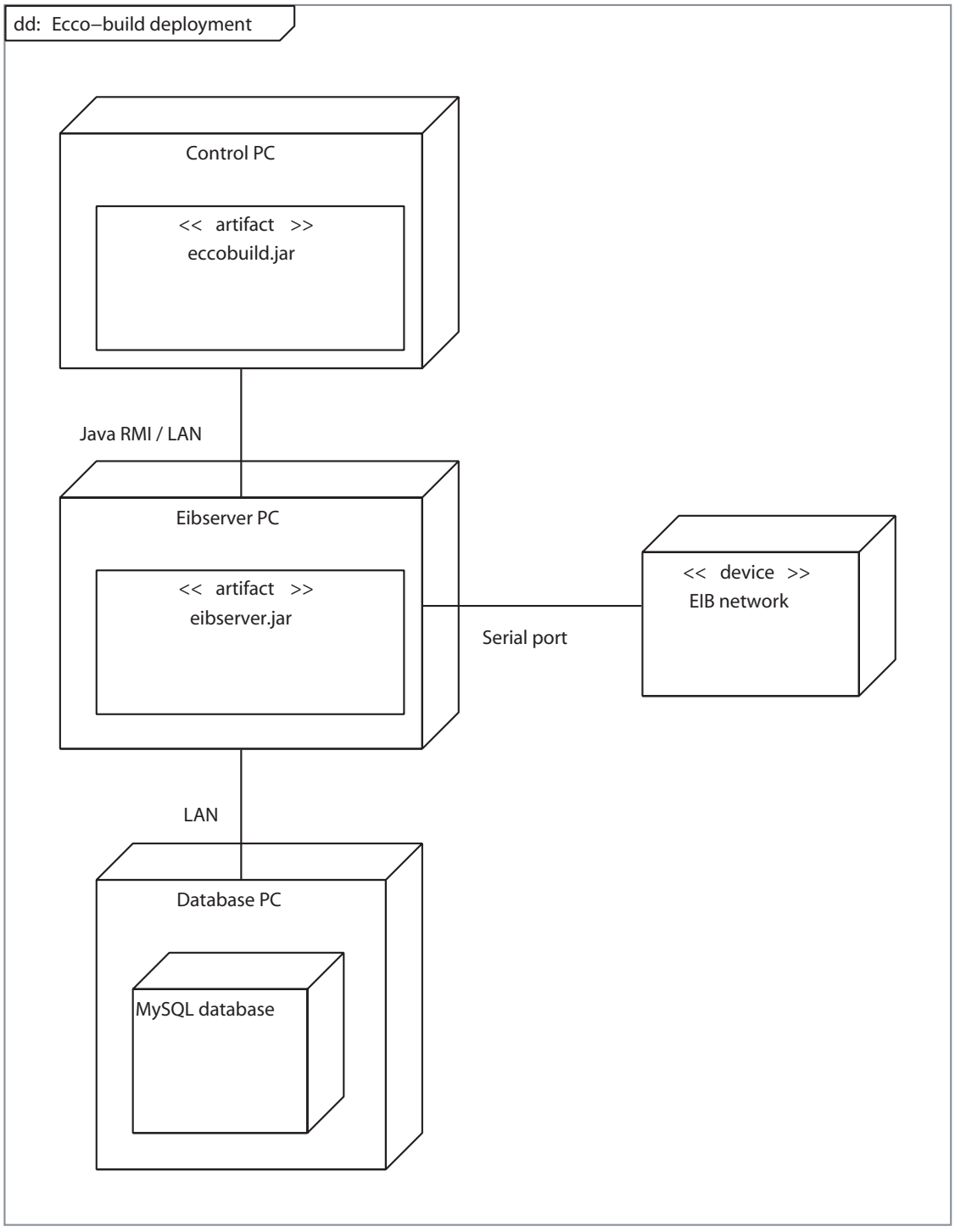


Figure 6.6: Controller deployment diagram. The control PC runs the executable `eccobuild.jar` file and communicates via Java RMI with the PC where EIBSERVER runs. This latter PC communicates via a serial port with the EIB network. This diagram also shows the separate PC on which a MySQL database runs and on in which all events are logged. `<<artifact>>` is the common UML term for physical manifestation of software, such as files. `<<device>>` denotes hardware, in this case the EIB.

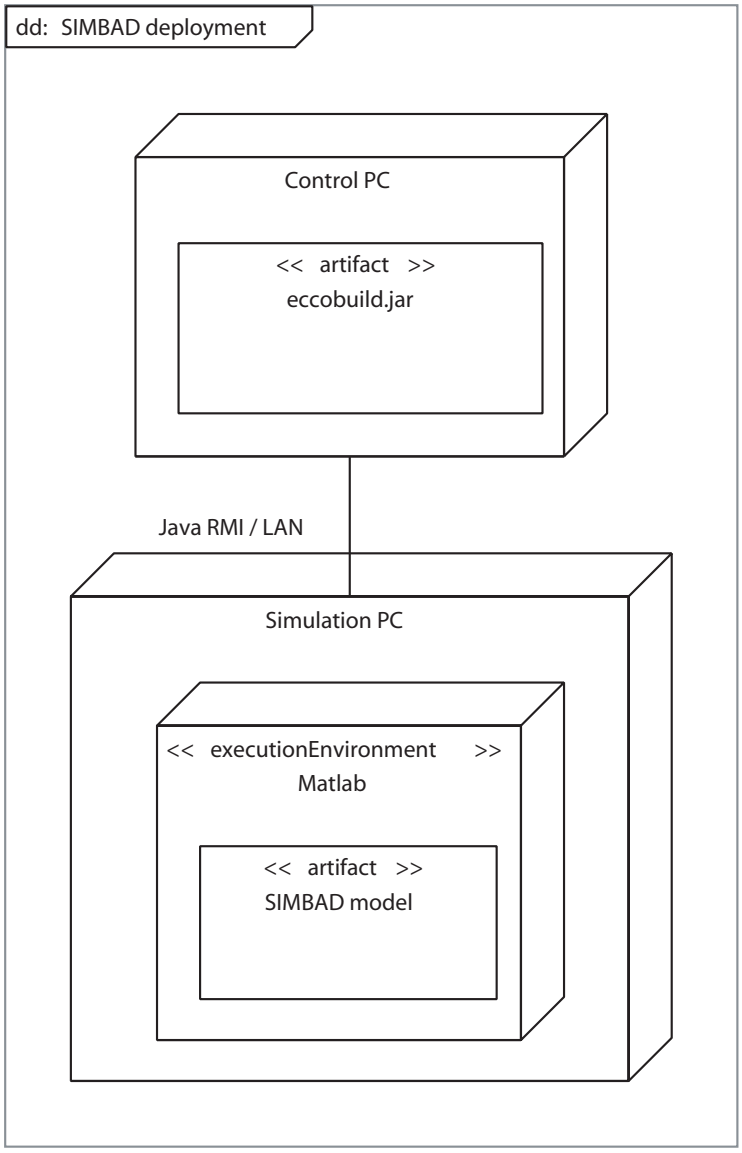


Figure 6.7: Deployment diagram when running against a simulation. The simulation PC runs the SIMBAD simulator under the MATLAB execution environment.

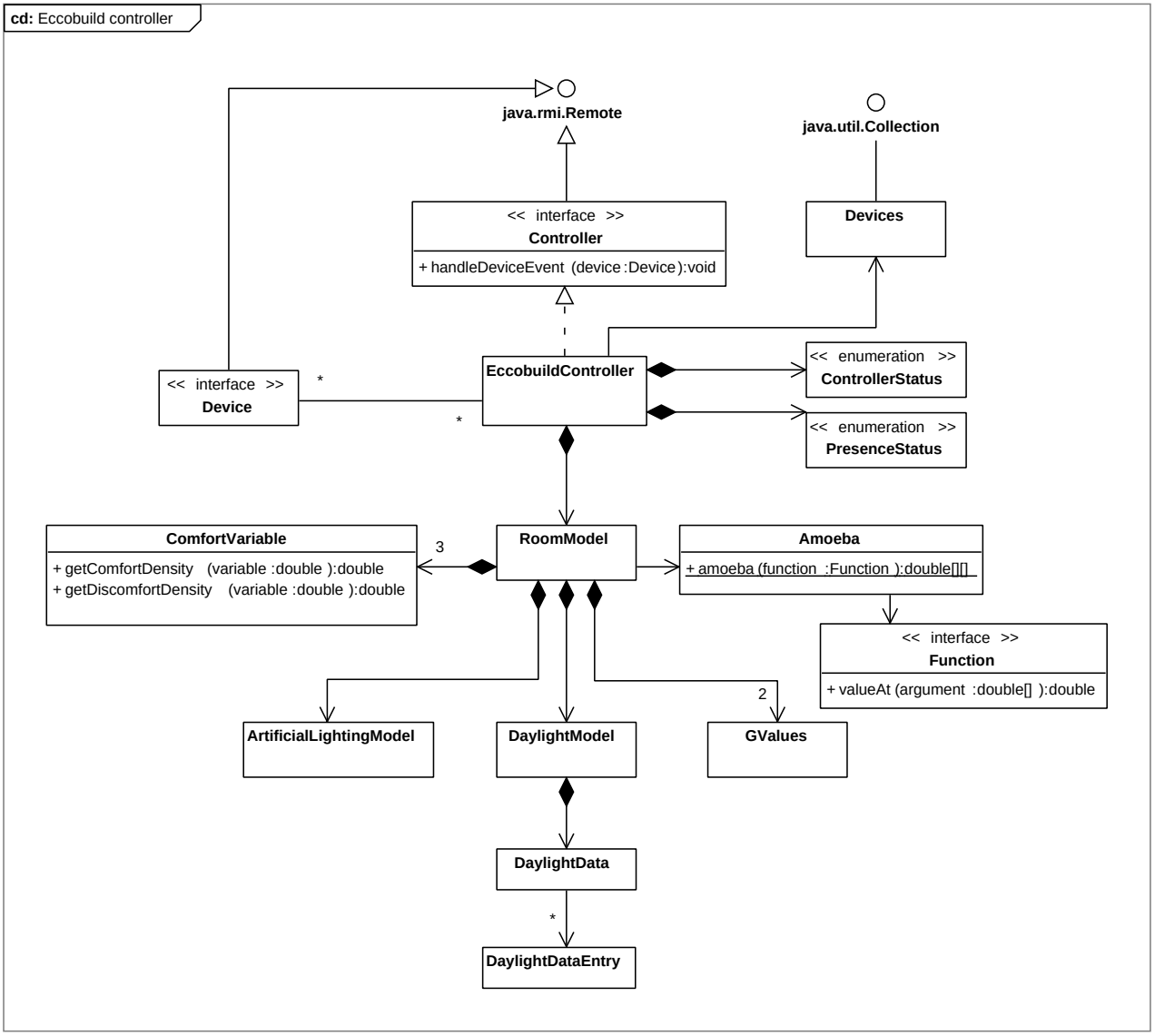


Figure 6.8: Controller class diagram. See the text for the description of each element. The arrows' exact meanings are defined by the UML standard, but always imply some sort of dependency of one class on the other.

`EccobuildController` receives event notifications from the remote devices and sends its own commands to them, but the software representation of the controlled zone is the responsibility of the `RoomModel` class. The instance of this class is a virtual model of the controlled room, whose devices our controller can experiment with until it finds the set of optimal commands.

`RoomModel` uses three instances of the `ComfortVariable` class, one for each variable that will enter the bayesian discomfort estimator whose equation was given by Equation (5.20). This latter class must therefore provide the density estimates of that variable for uncomfortable respectively comfortable situations. It does this through the `getDiscomfortDensity()` and `getComfortDensity()` functions. We have discussed in section 6.2.2 where these objects get their density estimates from.

`RoomModel` further uses an `ArtificialLightingModel` class for modeling the electric lighting (using the calibration curves mentioned in section 3.2.1), a `DaylightModel` class for modeling daylighting, and one instance of a `GValues` class per venetian blinds for modeling the solar gains, using the g-value data files described in section 3.2.5. We have discussed the implementation of the `DaylightModel` in section 6.2.3.

The optimization routine is implemented as a static `amoeba()` method of a `Amoeba` class, so called because of the optimization algorithm chosen for this work, and discussed in section 6.2.5. The main argument to this method is an object that implements the `Function` interface, the standard way of passing function-like arguments in Java.

The whole system is initialized when `EccobuildController` asks the singleton[8] `Devices` class for an enumeration of all available remote devices. `Devices` is responsible for establishing the initial RMI communication with the remote computer and for retrieving a list of all devices, their types and their names.

6.4.2 System initialization and typical event loop

The sequence diagram of Figure 6.9 shows how the controller initializes itself. It asks first for the (unique) instance of the `Devices` class. If this class has not been instantiated yet (as it never is before our controller initializes itself) it will use the `java.rmi.Naming` class to enumerate all available remote devices, and add a reference to each of them to a collection. `Devices` implements the `Collection` interface, so our controller next asks for an iterator over this collection of remote devices.

Each device is inspected in turn, and the controller subscribes to events on devices that are in the controlled room, or that monitor outdoor conditions.

Once the controller has finished registering on all devices it waits for events. It should not repeatedly query the remote devices for new events—this would be a waste of CPU time and network bandwidth. Instead, EIBSERVER provides a mechanism whereby remote objects can register to be notified of new events[9]. Our controller uses this mechanism. It registers for event notification through the `Device.addController()` method, passing a reference to itself as argument. When a sensor or actuator senses a change of state, is calls the `Controller.handleDeviceEvent()` method on the controller, passing itself back as argument. Our controller then dispatches the event to an appropriate handler method according to the type of the originating device[10].

[8] I.e., only one instance of that class can exist (Gamma et al., 1995, *Singleton* pattern).

[9] Also known as the *Observer* design pattern (Gamma et al., 1995, p. 293)

[10] This is done *asynchronously*, i.e. the `handleDeviceEvent()` method returns immediately after the event has

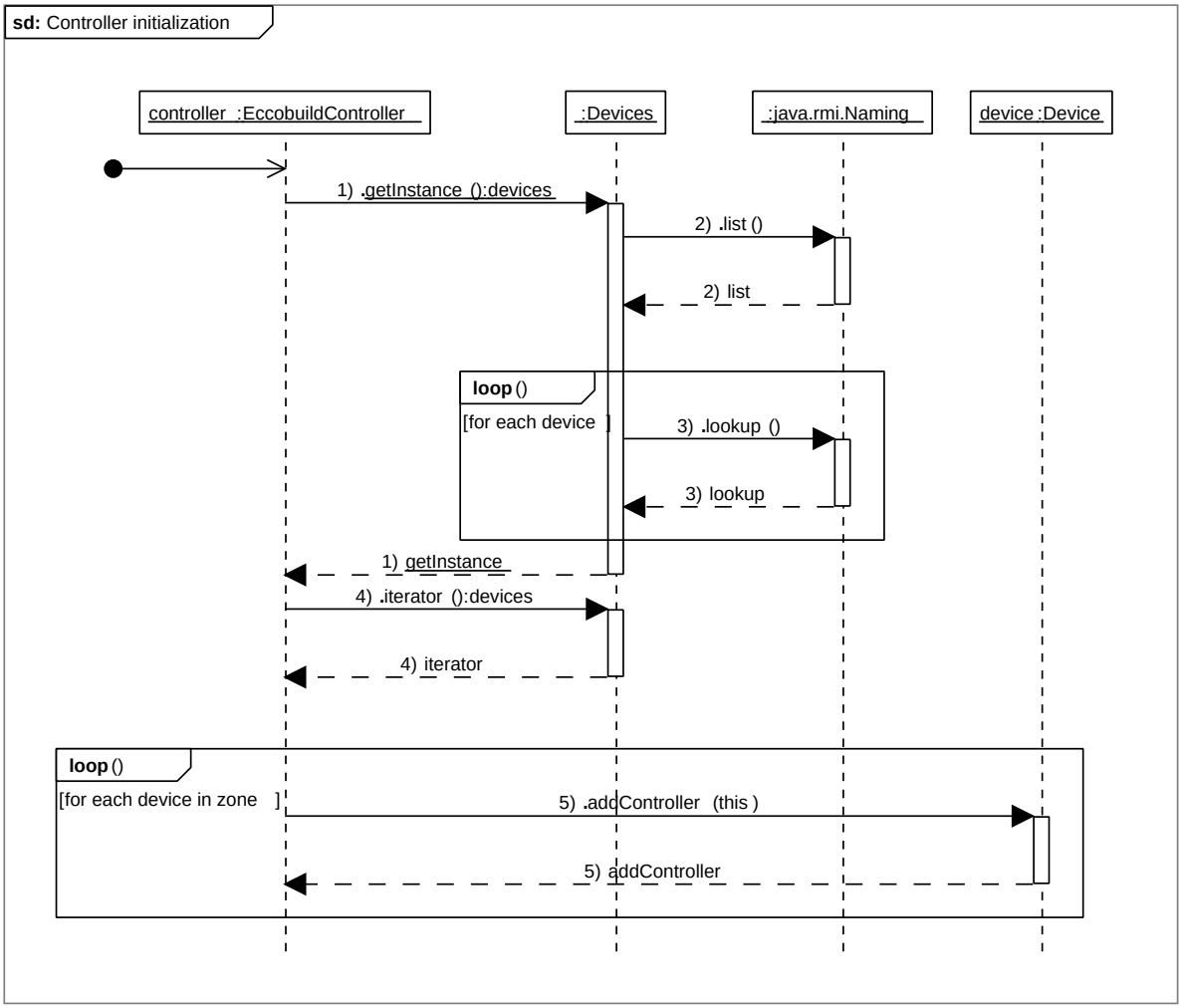


Figure 6.9: Controller initialization sequence diagram.

Thus, a new thread is created in the controller for each event, and is destroyed once the controller has finished handling this event. Between two events, the controller sits idle.

The BF3 sunshine sensor has been programmed to send an event at least once per minute, which regularly "wakes up" the system. The BFS sensor was chosen because variations in the outdoor irradiance are, together with changes in the room's occupancy, the events to which the controller must respond in a timely manner. We will now walk through such an event.

The sequence diagram on Figure 6.10 shows what happens when the controller receives an event on the sunshine sensor. First the `handleDeviceEvent()` method dispatches the event to the `handleSunChange()` method. The room model is updated with this new information. Then the controller enters its `checkForAction()` method, which contains all the logic necessary to decide whether to send new commands or not. It sets the room model to the current control values and asks the room model to evaluate the cost function's current value. If the user is present, the room model needs to know the illuminance distribution to answer that query. Let's assume the user is present.

We won't show on this diagram the details of the `cost()` method, but the general idea is the following. Since this is a new situation, no illuminances have been cached yet and the room model must ask the daylighting model to model the indoor daylighting illuminance. The room model also adds to this illuminance the contribution from the electric lighting. Then the three instances of the `ComfortVariable` class provide the density estimates necessary for computing the bayesian discomfort estimation. Finally, the energy cost for the electric lighting, and the thermal one associated with the solar gains are added.

The controller then asks the room model to try and find a set of control values that minimizes the cost function. The room model delegates the minimization algorithm itself to the `Amoeba` class and provides it with a reference to the cost function. `Amoeba` repeatedly evaluates the cost function for different control values, each of which necessitates a call to the daylighting and the electric lighting models.

When the algorithm has converged, the best control values and their corresponding cost function value are returned to the controller, who then filters these values through an output filter to prevent unnecessarily small blinds movements. After passing through the filter, the `sendValues()` method is called to send the control values to each corresponding device.

6.4.3 Controller states

The controller must not send commands too often, in order not to annoy the user. It must also recognize when the user overrides the controls and not intervene. In parallel, the controller must also track the user's presence status. It cannot blindly rely on the occupancy sensor, and for two reasons. First, some presence sensors incorrectly record as absent a user that sits very still, for instance while reading. Second, users that briefly exit the zone, for instance to refresh themselves, should not be considered by the controller as absent.

These requirements are solved by having the controller track two orthogonal states: the controller state, and the user presence state. Figure 6.11 shows the controller's state diagram. The most common state for the controller is the Automatic state, in which the controller listens to device events. After the controller sends a set of commands, it enters a Sleep state

been dispatched to a handler method. This handler method runs in its own thread. Otherwise, EIBSERVER would have to wait for our controller to finish processing each event before sending new ones. When running against the simulator, however, events are handled synchronously because our controller *must* finish processing each event before new ones arrive.

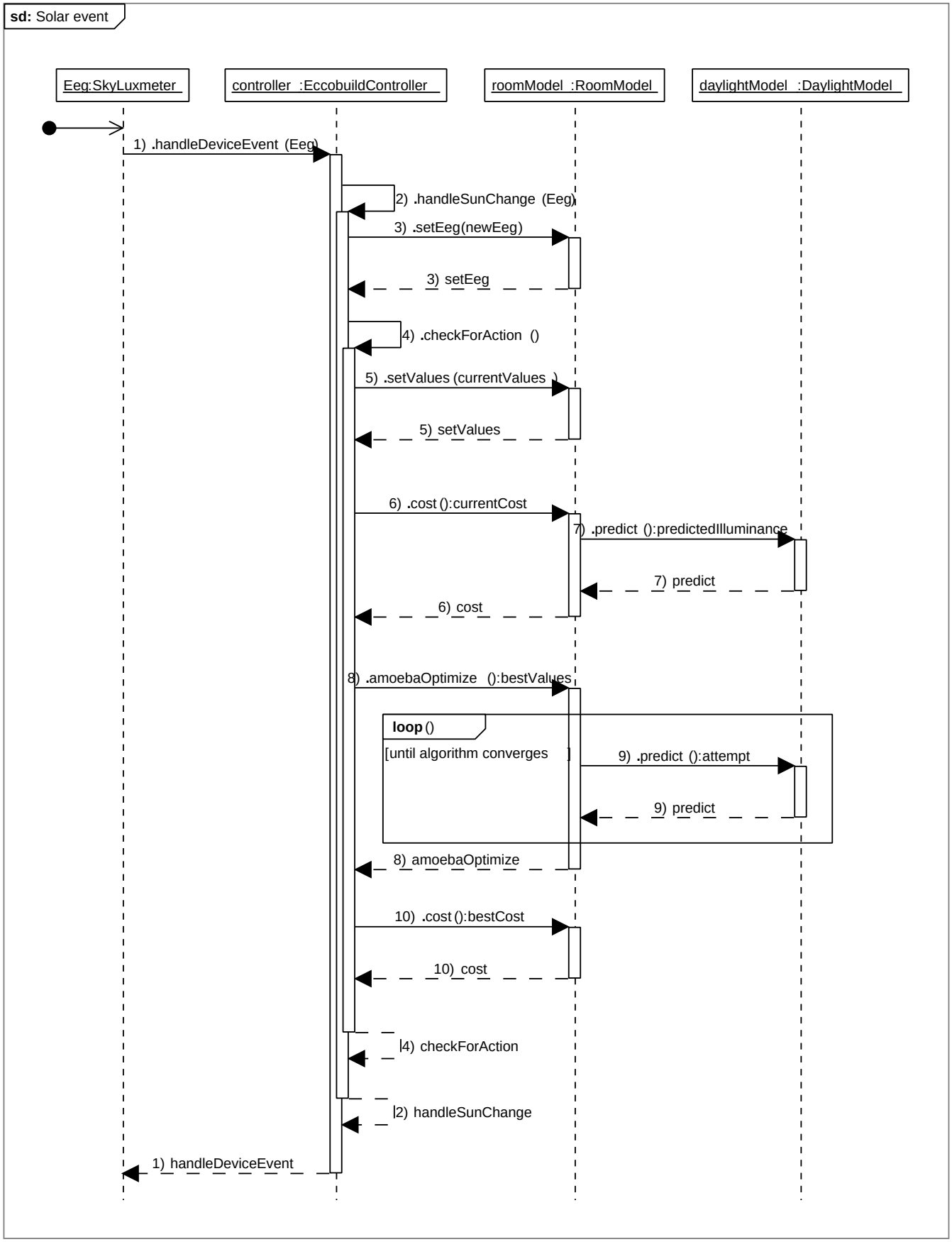


Figure 6.10: BF3 sunshine sensor event sequence diagram.

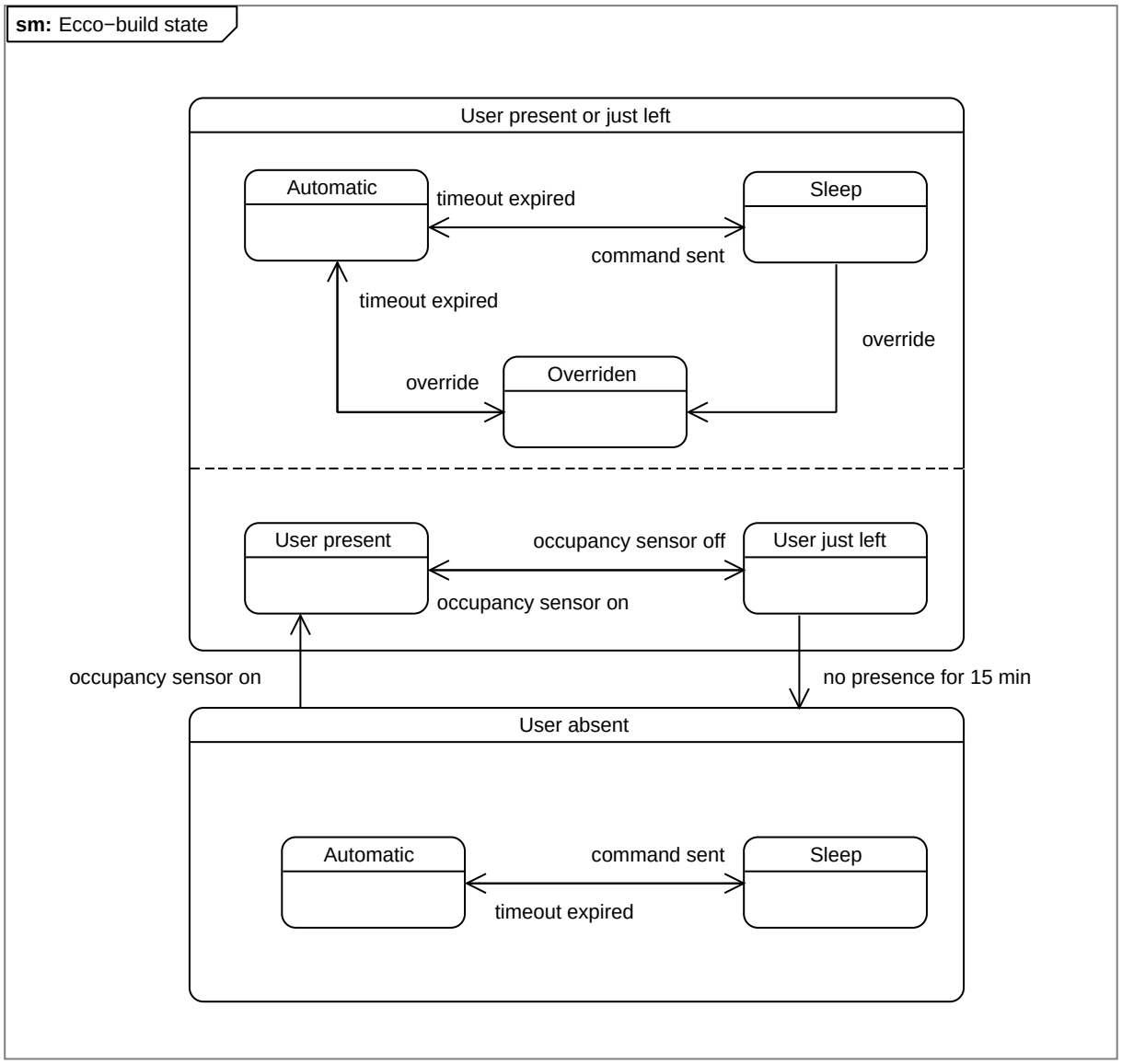


Figure 6.11: Controller state diagram.

and does not issue any other commands for 15 min, after which it returns to Automatic. If at any time the user overrides the controls, the controller enters the Overriden state and remains there for one hour or until the user is absent, whichever comes first.

Under exceptional conditions, the controller can send commands even when in the Sleep state. If, while sleeping, the controller detects that the situation can be significantly improved by lowering the blinds, then it will do so because this is interpreted as a significant glare risk that must be acted upon before waiting for the timeout to expire.

The user presence status is Present when the occupancy sensor detects the user as present. When the sensor does not detect the user anymore, the presence status enters a temporary Just Left state. If after 15 min the user has still not been detected, the presence status becomes Absent. While the user is absent the controller continues commuting between Automatic and Sleep states, but obviously cannot enter the Overriden state.

6.5 Chapter summary

In this chapter we have described the design of our controller and how it satisfies the controller's requirements. We have discussed how the control algorithm integrates the bayesian discomfort estimation algorithm and our simplified daylighting model, that together are the core of this work.

Besides the overall design, we have also focused on more detailed non-trivial design decisions, and briefly discussed alternative possible uses of the core tools developed in this work.

7 Controller tests

If the goal is to show the data, then show the data.

(William S. Cleveland)

In this chapter we present the results obtained from running our controller on-site on a real, occupied office room on one hand, and by computer simulation on several different virtual office rooms on the other hand.

The final version of our controller drives the venetian blinds and the electric lighting in an office room of the LESO building (room 201) since January 2007. We have also let it run against the different virtual office rooms described in section 3.2. We have analyzed the data acquired on these real and virtual office rooms and verified that our controller fulfills our requirements.

In section 7.1 we review the simulation runs that have been carried out within this thesis. In section 7.2 we examine first how the controller has adapted itself to the visual preferences of the occupant in office room 201. The parameters of this user-adaptation have been used when controlling the virtual office rooms, and we show how this adaptation has affected the illuminance distribution when our controller runs in these rooms.

In section 7.3 we analyze the energy demands in office room 201 and in the computer-simulated virtual office rooms, and examine the energy savings made possible with our controller.

7.1 Control runs on virtual and real office rooms

Most of the controller development happened in 2005 and 2006, and was much helped by the availability of the Simbad simulation model (c.f. section 3.2). Preliminary releases of the controller software ran intermittently in early 2006 on office rooms 201 and 202 but were interrupted between July and early 2007. A far improved version was then launched on office room 201 (office room 202 is now used for another project).

Several simulation runs were carried out during the controller's development, each of which helped identify and fix trouble areas. A comprehensive list of all simulations is given in Table 7.1. Only when the controller was running according to its requirements did we carry out a computer simulation for each combination of office room location (Rome or Brussels), orientation (north, west or south), and mode of operation (manual or controller). These full simulations are the last thirteen rows of Table 7.1.

We monitored the progress of the controller's quality by keeping track of open, closed and total number of issues (or "bugs", but an issue can also mean an efficiency improvement). The evolution of open and total number of issues since December 2005, when issue tracking started, is shown in Figure 7.1. The interruption of the software's development in the second half of 2006 is evident. It resumed in early 2007, aided by the simulation model. No more issues have been opened since April 2007, and we therefore believe the controller software is

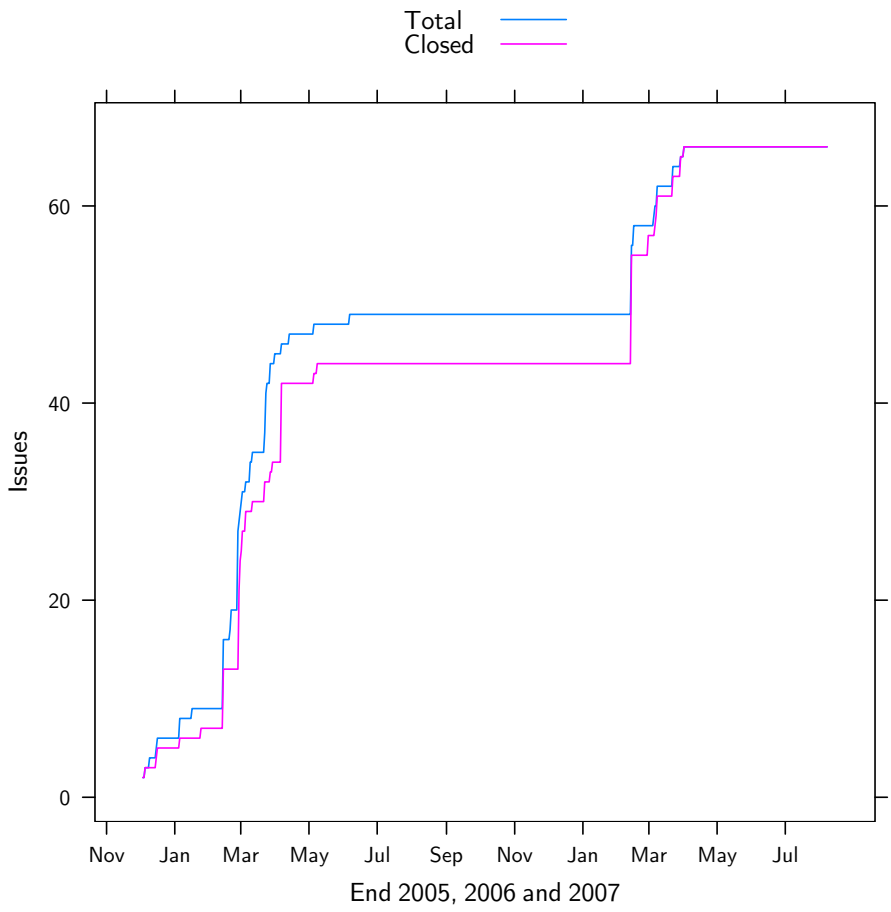


Figure 7.1: Total (open + closed) and closed controller software issues over time.

now free of any obvious errors[1].

7.2 User visual comfort

7.2.1 Adaptation to a real user

The sole occupant of LESO office room 201 adjusted either the blinds or the electric lighting of his office room 551 times since the controller began monitoring the three illuminance sensors. Of these, 434 happened in the first half of 2006, or an average of 2.40 per day (or 16.69 per week). The controller was not running for the second half of 2006 and resumed in 2007, during which time 117 user actions (in three months) were recorded.

These records are the basis on which the controller adapts to the user. The illuminance distribution before and after user action and for each sensor location is shown on Figure 7.2.

[1] The user has reported a tendency by the controller to leave the blinds uncompletely open during overcast skies, about 80–90 % open. We believe these situations correspond to "shallow" minima of the cost function, i.e. the small potential improvement of the cost function does not warrant moving the blinds. We could have added a special rule for these situations, but deem such a measure to be clumsy. We believe that alternatives to a cost function should be explored instead, and do not consider this as a controller "bug".

Log file	Data file	Location	Orientation	Mode	Duration	Comments
2006-12-14T1740.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-20T1605.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-20T1645.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-20T1655.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-21T0840.log	2006-12-21T0840.mat	Brussels	South	Controller	1 month	First run w/o errors
2006-12-21T1340.log	N/A	Brussels	South	Controller	< 1 week	Fixed convergence failures
2006-12-21T1420.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-21T1440.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-21T1510.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-21T1520.log	N/A	Brussels	South	Controller	< 1 week	
2006-12-21T1600.log	2006-12-21T1600.mat	Brussels	South	Controller	1 month	
2006-12-22T1200.log	2006-12-22T1200.mat	Brussels	South	Controller	1 month	
2007-01-03T1630.log	2007-01-03T1630.mat	Brussels	South	Controller	1 month	Was not committed to repository
2007-01-04T0910.log	2007-01-04T0910.mat	Brussels	South	Controller	1 month	Eeg feedback every 5 min
2007-01-05T1100.log	2007-01-05T1100.mat	Brussels	South	Controller	1 month	
2007-01-08T1700.log	2007-01-08T1700.mat	Brussels	South	Controller	1 year	First succesful year-long run
2007-01-26T1940.log	2007-01-26T1940.mat	Brussels	South	Controller	1 month	
2007-01-28T2000.log	2007-01-28T2000.mat	Brussels	South	Controller	1 year	
lost	2007-02-28T2000.mat	Brussels	South	Controller	1 year	Detect unnecessary weak lighting
2007-03-07T1045.log	2007-03-07T1045.mat	Brussels	South	Controller	6 months	
2007-03-09T1500.log	2007-03-09T1500.mat	Brussels	South	Controller	1 year	
2007-03-11T1800.log	2007-03-11T1800.mat	Brussels	South	Controller	1 year	
2007-03-15T1000.log	2007-03-15T1000.mat	Brussels	South	Controller	1 year	Thermal costs depend on user presence
2007-03-22T1520.log	2007-03-22T1520.mat	Brussels	South	Controller	1 year	Simulator gives no blinds feedback
2007-03-27T1015.log	2007-03-27T1015.mat	Brussels	South	Controller	1 year	Simplified thermal cost term
2007-03-29T1730.log	2007-03-29T1730.mat	Brussels	South	Controller	1 year	Fixed exceptions in daylight model
2007-04-01T1630.log	2007-04-01T1630.mat	Brussels	South	Controller	1 year	Ensure lights off when user absent
2007-04-02T1700.log	2007-04-02T1700.mat	Rome	South	Controller	1 year	
2007-04-05T1500.log	2007-04-05T1500.mat	Rome	West	Controller	1 year	
2007-04-09T1200.log	2007-04-09T1200.mat	Rome	North	Controller	1 year	
2007-04-10T1410.log	2007-04-10T1410.mat	Brussels	North	Controller	1 year	
2007-04-12T0900.log	2007-04-12T0900.mat	Brussels	South	Controller	1 year	No initial daylighting data
2007-04-18T1500.log	2007-04-18T1500.mat	Brussels	West	Controller	1 year	
N/A	YearEvaluation_BrusselNorth_HBD_VB.mat	Brussels	North	Manual	1 year	
N/A	YearEvaluation_BrusselSouth_HBD_VB.mat	Brussels	South	Manual	1 year	
N/A	YearEvaluation_BrusselWest_HBD_VB.mat	Brussels	West	Manual	1 year	
N/A	YearEvaluation_RomeNorth_HBD_VB.mat	Rome	North	Manual	1 year	
N/A	YearEvaluation_RomeSouth_HBD_VB.mat	Rome	South	Manual	1 year	
N/A	YearEvaluation_RomeWest_HBD_VB.mat	Rome	West	Manual	1 year	

Table 7.1: SIMBAD simulation runs.

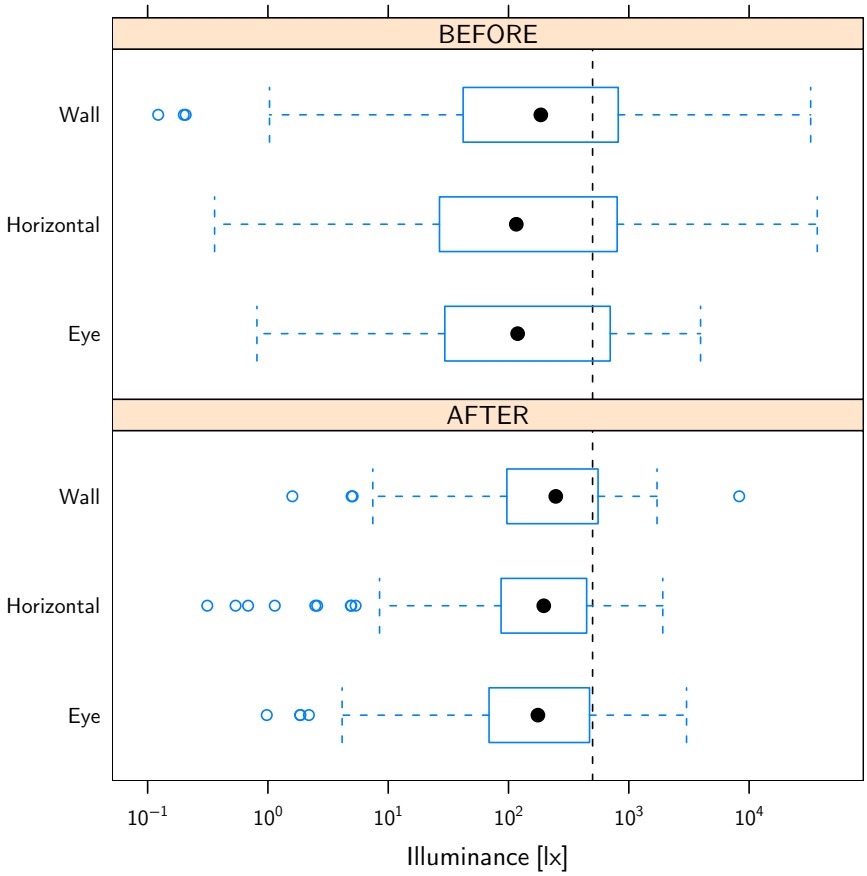


Figure 7.2: Illuminance distribution before and after user action in office room 201. The dashed line is at 500 lx.

The range of illuminances that trigger an user action is very wide, whereas the resulting ones are concentrated in a narrower range, bearing out the hypothesis that the illuminances at these sensor locations correlate with visual discomfort.

It is, however, difficult to assess the illuminance density distribution from this figure. Figure 7.3 remedies this by computing the taut-string density estimate of each distribution. These plots confirm that the range of post-action illuminances is much narrower than for pre-action ones, and that their location is shifted.

These density estimates are those we need in order to apply the bayesian discomfort estimation from Equation (5.20). From our data, we first derive the discomfort probability as a function of each of the three illuminance measurements alone. These probability estimates are shown in Figure 7.4. These curves show that a global minimum of the estimated discomfort probability is reached close to 500 lx for both the wall- and eye-level sensors, and perhaps slightly lower for the horizontal workplane sensor.

The density estimates shown in Figure 7.3, however, allow us also to compute the total bayesian discomfort probability through Equation (5.20). This equation computes the probability as a function of three illuminances—a better estimate of the user's visual discomfort

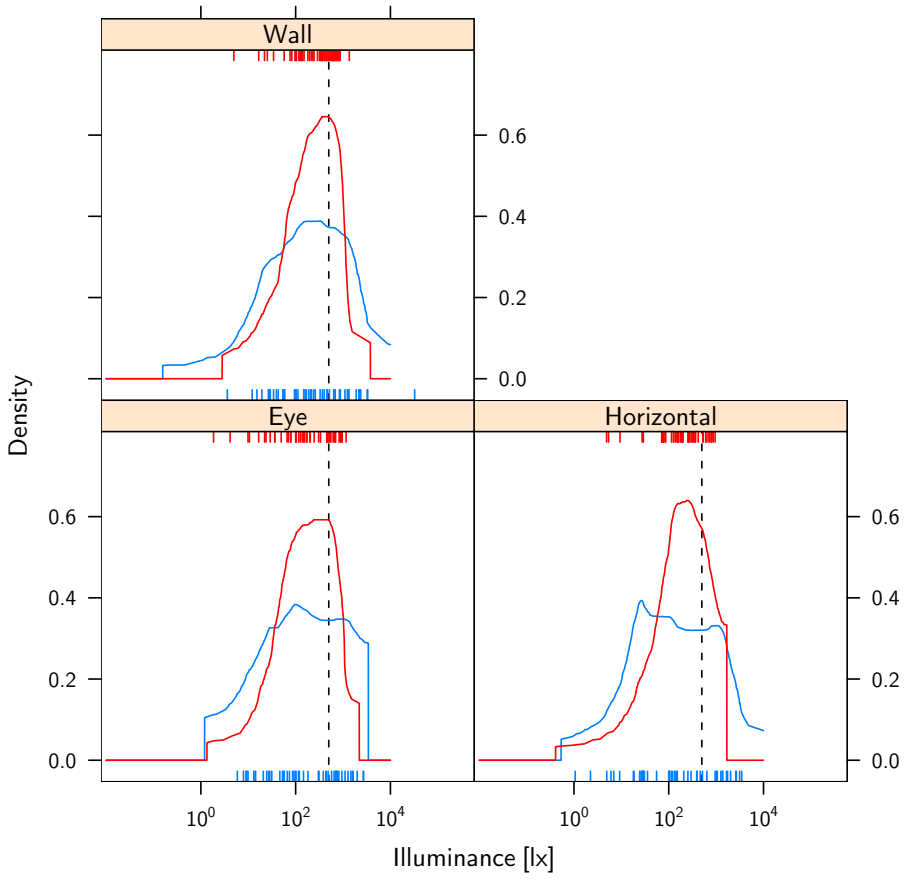


Figure 7.3: Density estimates of pre- and post-action illuminances in office room 201. Post-action illuminances are shown in red. 2×50 randomly chosen illuminance values are shown as ticks on each panel. The dashed line is at 500 lx.

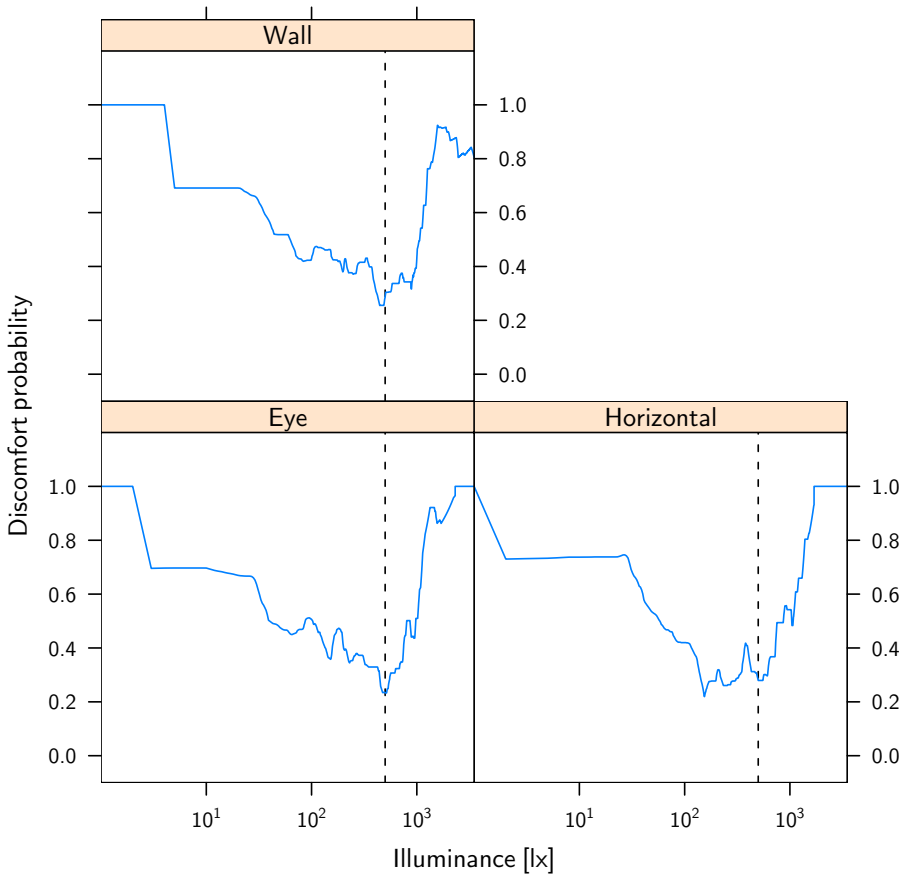


Figure 7.4: Discomfort probability estimates on office room 201, for individual illuminance sensors. The dashed line is at 500 lx.

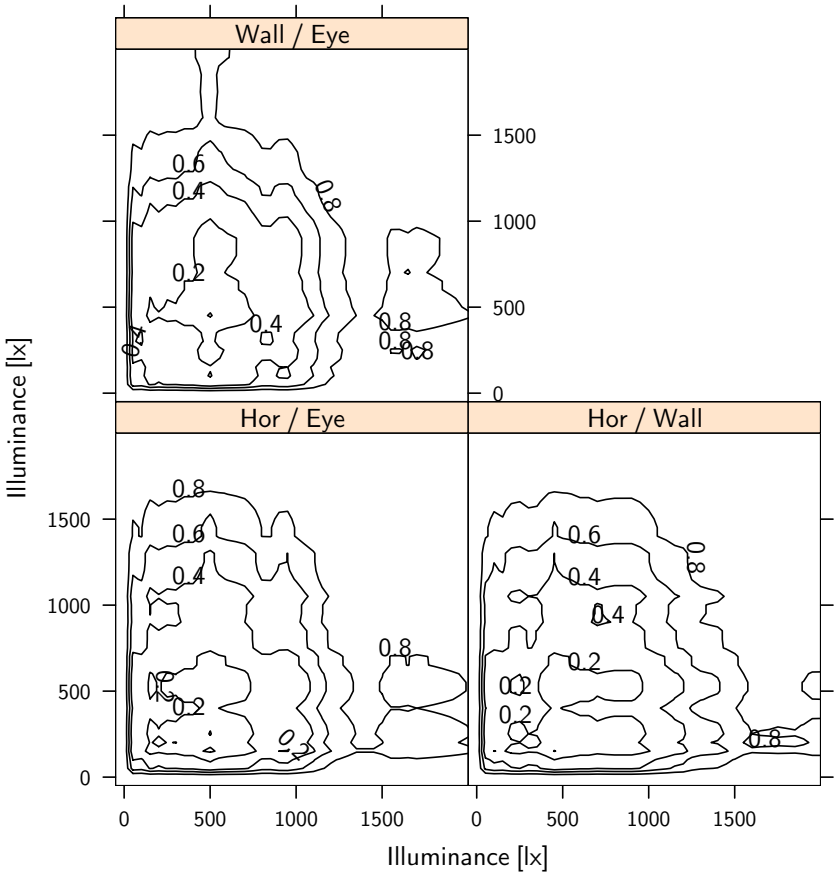


Figure 7.5: Discomfort probability estimates from pairs of comfort variables. The first variable in the strip is on the vertical axis, the second on the hozirontal one.

than if we had only one of the individual curves shown in Figure 7.4.

This probability is a function of three variables. Before we attempt to visualize it we show first in Figure 7.5 the estimated discomfort probabilities obtained from taking the illuminances values in pairs, i.e. the results of applying Equation (5.20) with only two comfort variables. These plots show that the eye-level illuminance has the largest influence on visual discomfort: the rate of probability variation is highest when it is the eye-level illuminance that varies. This was already suggested by Figure 7.4. The wall illuminance has the least influence on visual discomfort. These plots suggest also that the minimum visual discomfort would be obtained in situations whose resulting horizontal illuminance is between 250–500 lx and the eye-level illuminance close to 500 lx. The resulting wall illuminance should not be higher than 1000 lx nor lower than about 150 lx.

We finally show in Figure 7.6 slices taken at 50 lx intervals along the wall illuminance dimension in the three-dimensional matrix that holds the probability estimates for each triplet of illuminances. The global minimum, 0.0320, is found for an eye-level illuminance of 500 lx,

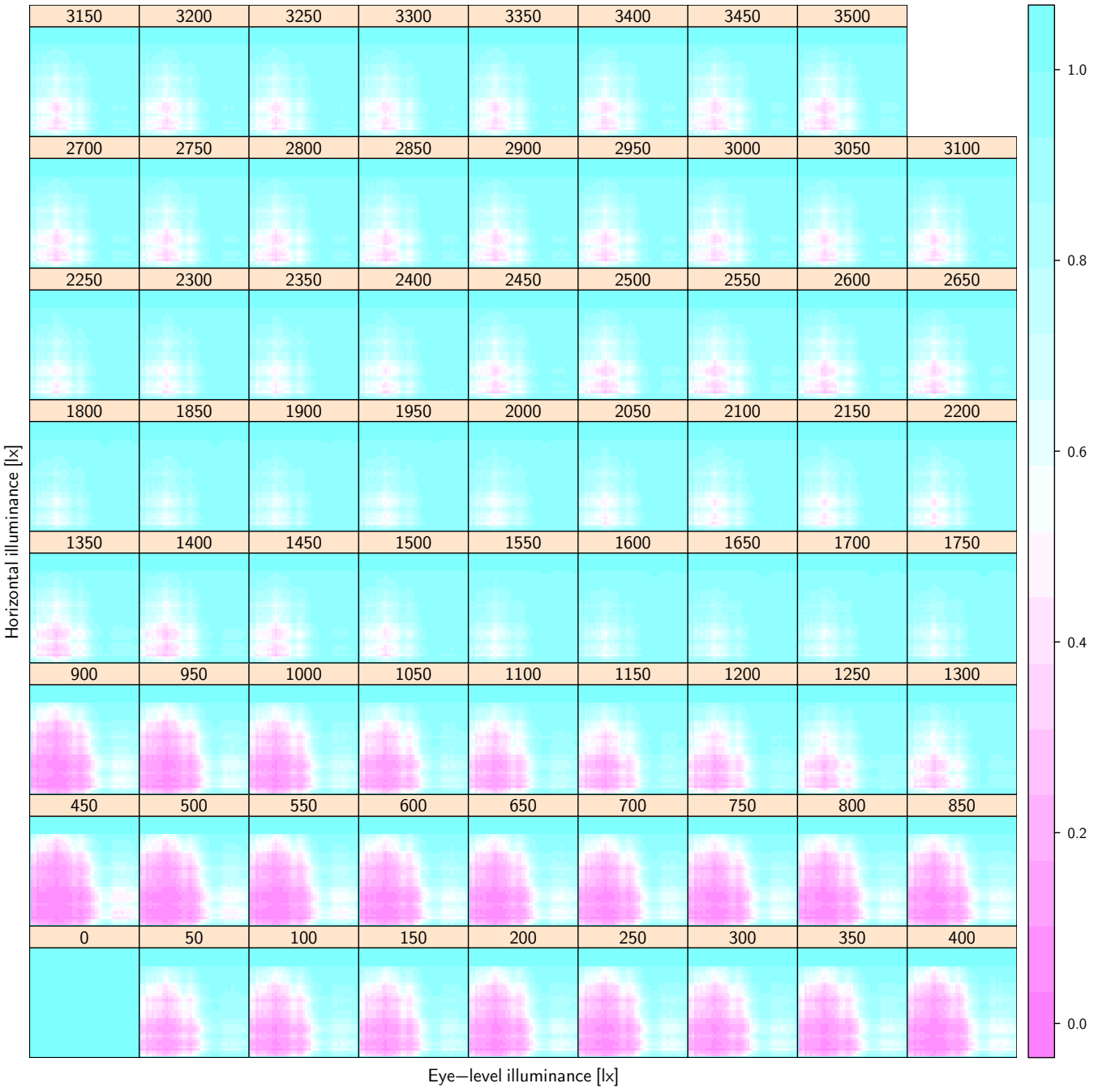


Figure 7.6: Discomfort probability estimate as a function of eye-level, horizontal and wall illuminance. Each panel is a constant wall illuminance slice in a three-dimensional matrix that holds the probability estimate for each triplet of illuminances. The wall illuminance is given in each panel. The horizontal and eye-level range of illuminance is 0–2000 lx.

an horizontal illuminance of 250 lx, and a wall illuminance of 450 lx[2]. If we fix arbitrary limits to the visual discomfort probability at 0.5, this plot suggests that in office room 201, the wall illuminance should not exceed 1200–1300 lx, the eye-level illuminance should not exceed 700–800 lx, and the horizontal illuminance should not exceed 1000 lx.

Taken together, these results illustrate how on the basis of recorded user actions alone our algorithm computes the estimated visual discomfort as a function of the illuminance measured on three points in the office room. It is this function that is used internally by the algorithm when it explores the blinds' settings and tries to optimize the cost function.

To conclude this section, we test now whether the number of user overrides has significantly changed from 2006 to 2007, from manual to controller mode.

Between 1 February and 1 May 2006, there were 233 user actions. In 2007 there were 91. The user was present in his office 390.60 hours during these three months in 2006, and 321.34 hours in 2007. The number of user actions in 2007, normalized to the user occupancy in 2006, is thus 110.61.

Let A and B be the number of user actions in 2006 and 2007 respectively. If the controller has no effect on the number of user actions, then A and B will be drawn from the same Poisson distribution and $\frac{A-B}{\sqrt{A+B}}$ will be normally distributed with mean 0 and variance 1. There is thus a 99% probability of finding this value in the range $[-2.57; 2.57]$.

But here, $\frac{A-B}{\sqrt{A+B}} = -6.64$, and the hypothesis is thus rejected. We conclude that our controller has had a significant effect on the number of user interactions, which have been reduced almost by half.

7.2.2 Estimated visual discomfort in virtual office rooms

The horizontal illuminance distribution during the controller's first year of operation in the south-oriented Brussels virtual office room, without initial daylighting training data, is given in Figure 7.7. Timesteps when the electric illuminance was more than 10 lx have been filtered out in order to concentrate on daylighting illuminance alone. This figure shows also the illuminance distribution at the same timesteps in manual mode. These plots make it clear that even with no initial training data, the resulting illuminance distribution given by the controller is comparable to the one in manual mode (the user's behaviour model was described in section 3.2.2), strongly suggesting that the visual discomfort is minimized. The illuminance distribution after one year of training data is shown for comparison.

We examine now the illuminance provided in all virtual office rooms by the controller after one year of training data taken on weekends, and compare it with the illuminance chosen by our modeled user. The density estimates of the horizontal workplane illuminance distributions when the user is present are given in Figure 7.8. The illuminance distribution in manual mode is generally shifted to higher values, and the sharp density increase close to 500 lx corresponds to the point when the user switches on the electric lighting, according to our user model.

As we will see in section 7.3.2, the largest relative energy savings are achieved by our controller on the electric lighting, so it is surprising to see that our controller turns it on much more aggressively than the user: the illuminance is sometimes as low as the first reference line on Figure 7.8 (corresponding to 33% power, or 260 lx), whereas in manual mode it never

[2]From Figure 7.4, the minimum discomfort probability for each variable is about 0.25, implying a ratio of density estimates between pre- and post-action illuminances of about $1/0.25 - 1 = 3$ (c.f. Equation (5.7). The combined discomfort probability is given by Equation (5.20): $p = 1/(1 + 3^3)$, or approximately 0.036, which is close to our actual result.

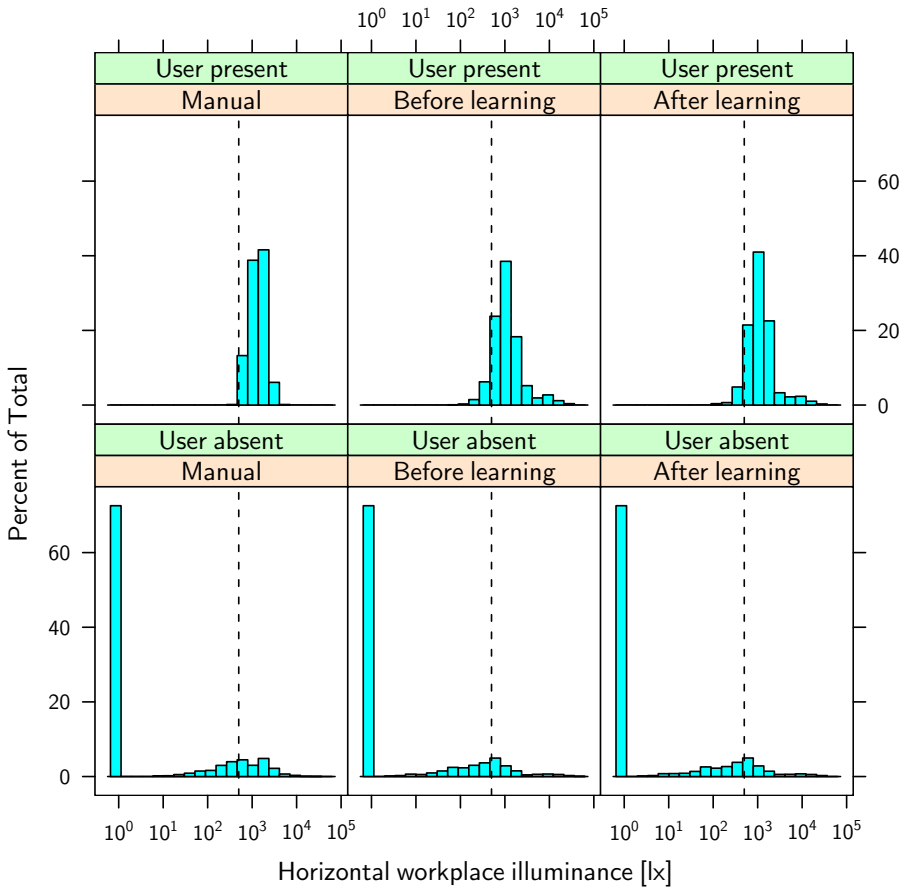


Figure 7.7: Horizontal daylight-only illuminance distribution in manual mode, in controller mode without initial learning, and in controller mode after one year of learning. The thin dashed line corresponds to 500 lx. Illuminances of zero lx have been binned in the first bin.

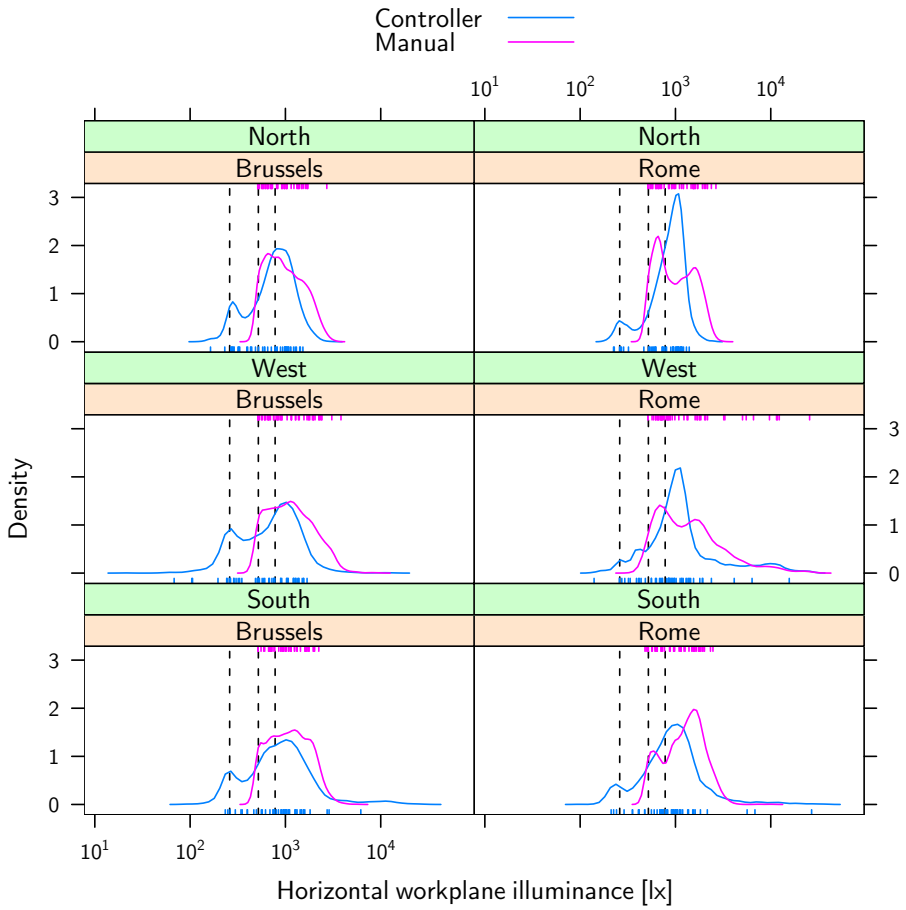


Figure 7.8: Horizontal workplane illuminance distribution in user presence. The electric lighting provided at 33%, 66% and 100% of full power (260, 520 and 780 lx) are shown as dashed lines. 2×50 randomly chosen illuminances are shown on each panel as tick marks.

goes below the 66% limit. This suggests that the controller begins to use the electric lighting at lower power than the user—and presumably more effectively.

From the wall, eye-level and horizontal workplane illuminances given by our simulation runs, we can now derive a bayesian discomfort estimate for an arbitrary user. We will do that for each timestep of each simulation run, using the data recorded on a real user in section 7.2.1.

Figure 7.9 shows how the bayesian discomfort estimates are distributed. Each panel shows the distributions for manual mode and controller mode for a given combination of office room location and location. The interquartile range in controller mode is in all cases much reduced, and the median is always lower than in manual mode. In controller mode, there are almost always many outliers but this is expected when there are so many data points (2080 hours of user presence per year). Note, however, that the whiskers of the controller mode's boxplots never extend to the full data range (except for Rome West), whereas they always do for the manual mode.

This does not necessarily suggest that the controller provides a more comfortable environ-

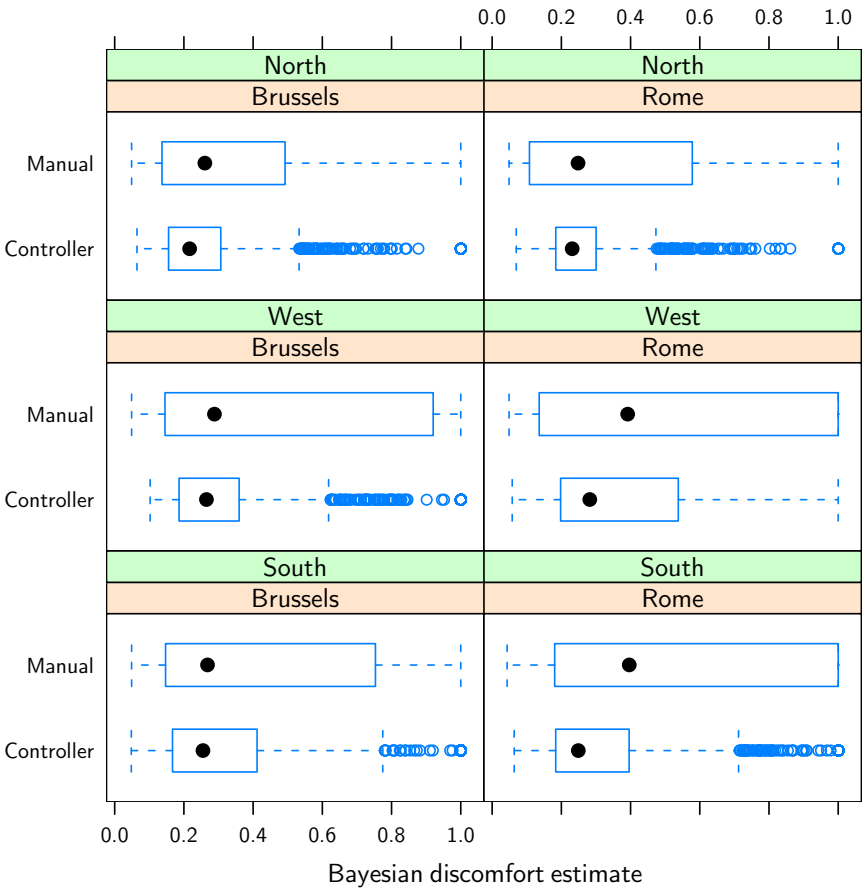


Figure 7.9: Bayesian discomfort estimate distribution in all virtual office rooms.

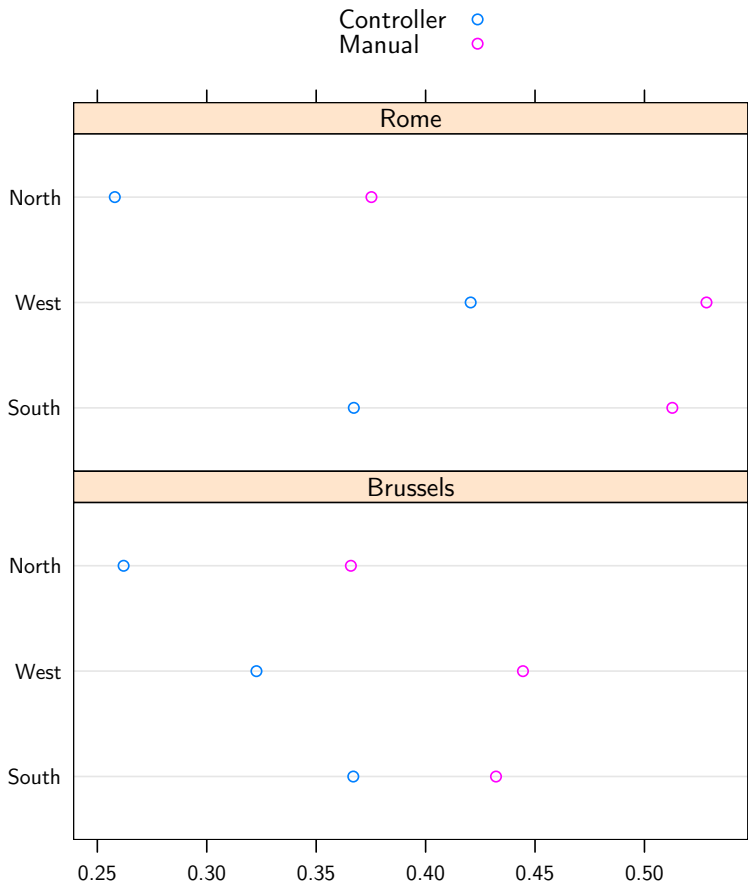


Figure 7.10: Average estimated visual discomfort in controller and manual mode, virtual offices.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Location	1	0.01	0.01	7.41	0.0297
Orientation	2	0.03	0.02	19.71	0.0013
Mode	1	0.04	0.04	45.06	0.0003
Residuals	7	0.01	0.00		

Table 7.2: Average yearly discomfort analysis of variance table.

ment than the real user could have provided. It merely suggests that the controller provides a more comfortable environment than would be obtained by an automatic controller programmed to behave as our simulated user.

We show in Figure 7.10 the average bayesian discomfort estimate for the virtual office rooms, in controller and manual modes. The average estimates are decreased by about 0.1 in every situation when using the controller. Ignoring any possible interaction between the variables (we do not have enough data to detect any such interaction), we can model the average discomfort estimate as a linear combination of the 'location', 'orientation' and 'mode' discrete variables (or *factors*). The best-fit coefficient of the 'mode' factor is then 0.11. In other words, switching from manual to controller mode decreases the visual discomfort probability on average by 0.11. It is the factor with the most influence among the three.

Table 7.2 gives the analysis of variance table for the average estimated discomfort. This analysis shows that location, orientation and mode all significantly affect the visual discomfort (their p-value is lower than 5%), but the single most significant factor remains the presence of our bayesian controller.

This study should *not* be taken to mean that the controller will satisfy 11 more users out of 100 than if we left them manage their environment manually. What is *does* mean is that our controller will decrease our user's average discomfort probability by 0.11, reducing it from 0.44 in manual mode to 0.33 in controller mode.

7.3 Energy performance

7.3.1 LESO office rooms 201 and 202

We show in Figure 7.11 the instantaneous power measured on the appliance[3] and heating meters in LESO office rooms 201 and 202. The appliance power is regular in both rooms. There is clearly a regular demand for about 50 W in room 201, likely for the user's laptop and monitor. An extra demand for about 150 W is periodically added to this base demand, which corresponds probably to the room's electric lighting.

Something in office room 201 draws about 10 W continuously and is recorded on the appliance meter. It was switched off between the third week of November 2006 and the second week of January 2007. Figure 7.12 shows how the power demand in office room 201 evolved during that time.

The appliance power in office room 202 is less predictable. There is a base power demand of almost 50 W corresponding to the office room's computer, but that computer is less often used than in office room 201. Indeed, that room is very irregularly occupied, but on those occasions its user seemed to systematically leave the computer switched on for days. Notice in

[3]The lighting power demand is measured on the same meter as the other appliances.

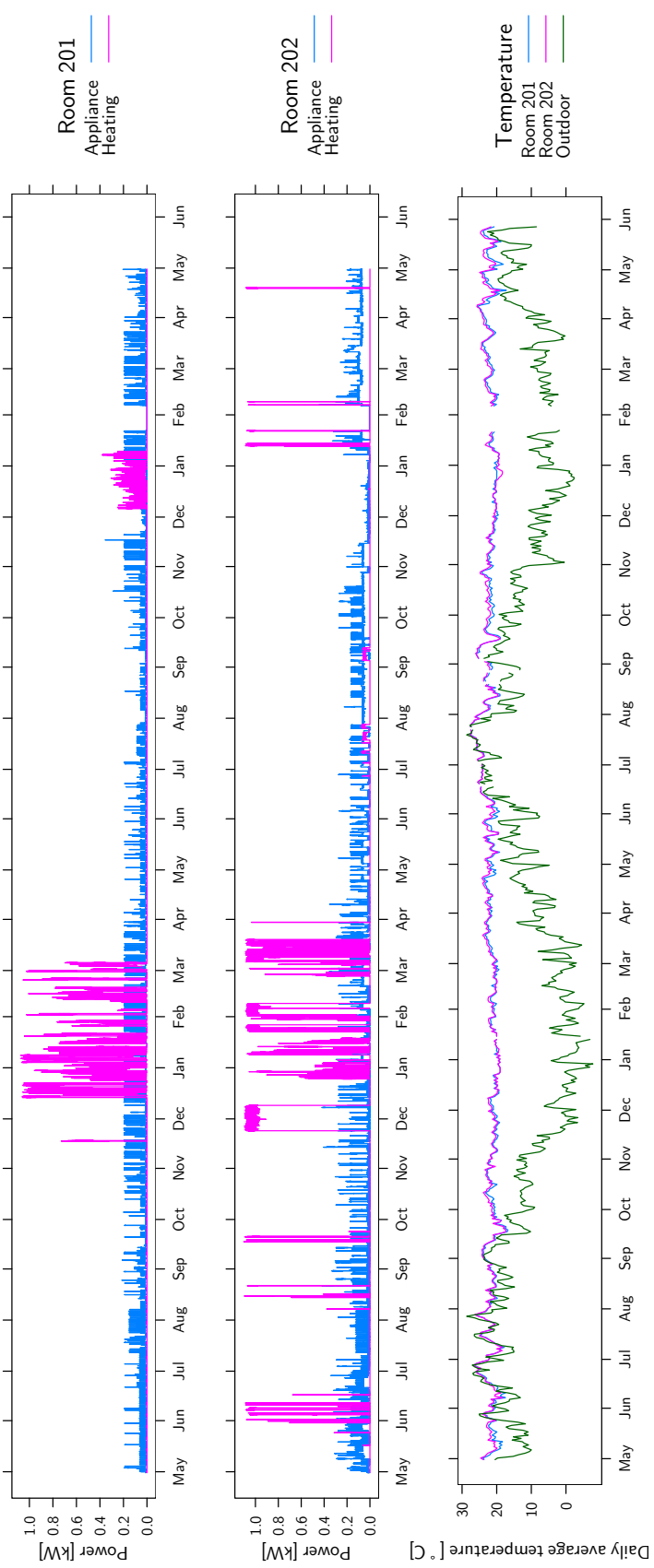


Figure 7.11: Electrical power demand in office rooms 201 and 202, indoor and outdoor daily average temperatures, 2005–2007. The relative uncertainty on the power demand is estimated at 8% for the heating and at 2% for the appliances.

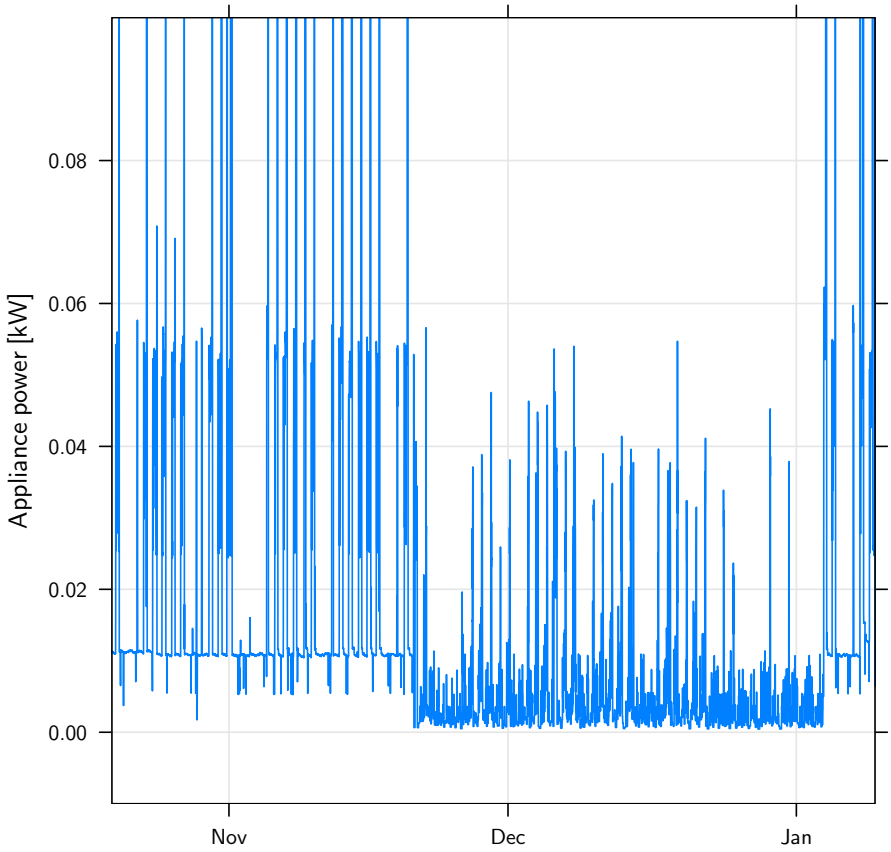


Figure 7.12: Lighting and appliance power demand in office room 201, end 2006 and beginning of 2007.

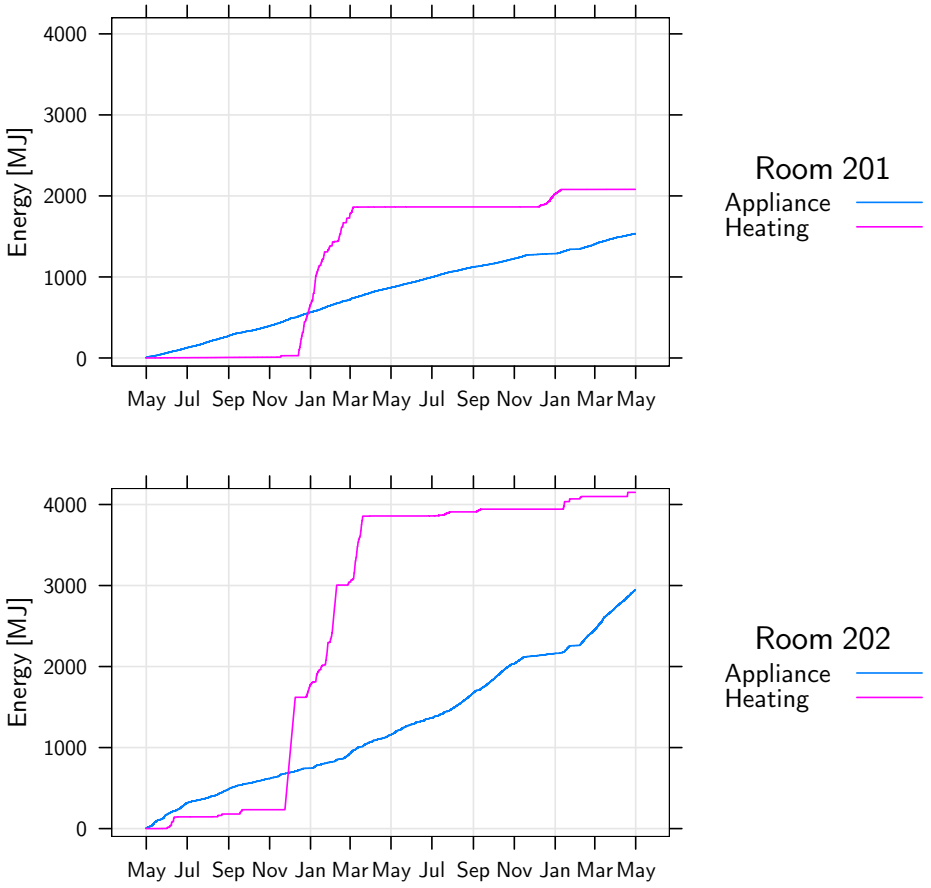


Figure 7.13: Cumulative power demand in office rooms 201 and 202. The relative uncertainty on the power demand is estimated at 8% for the heating and at 2% for the appliances.

particular the pattern that began in August 2006, when a base consumption of almost 50 W is almost continuously present.

The heating power demand in office room 201 is null outside of the coldest winter months. That office room's user is indeed known to keep the office room's thermostat to the absolute minimum (16 °C), and the power pattern we observe is not surprising.

That of office room 202 is much more erratic. The heating demands in June, August and September 2005 are difficult to understand, and the uninterrupted heating that lasted a week in early December 2005 baffles us. We have no explanation for these episodes—perhaps some other experiment was being carried out during these months in that room.

The cumulative energy consumptions of both office rooms are shown in Figure 7.13. The appliance consumption in office room 201 is very regular, about 800 MJ/year, and does not seem to have changed by the introduction of our controller. The removal of the 10 W device at the end of 2006 is just detectable. Notice, also, the dramatic difference in heating demand between the 2005–2006 and 2006–2007 winter seasons, attributable to the much milder 2006–2007 winter.

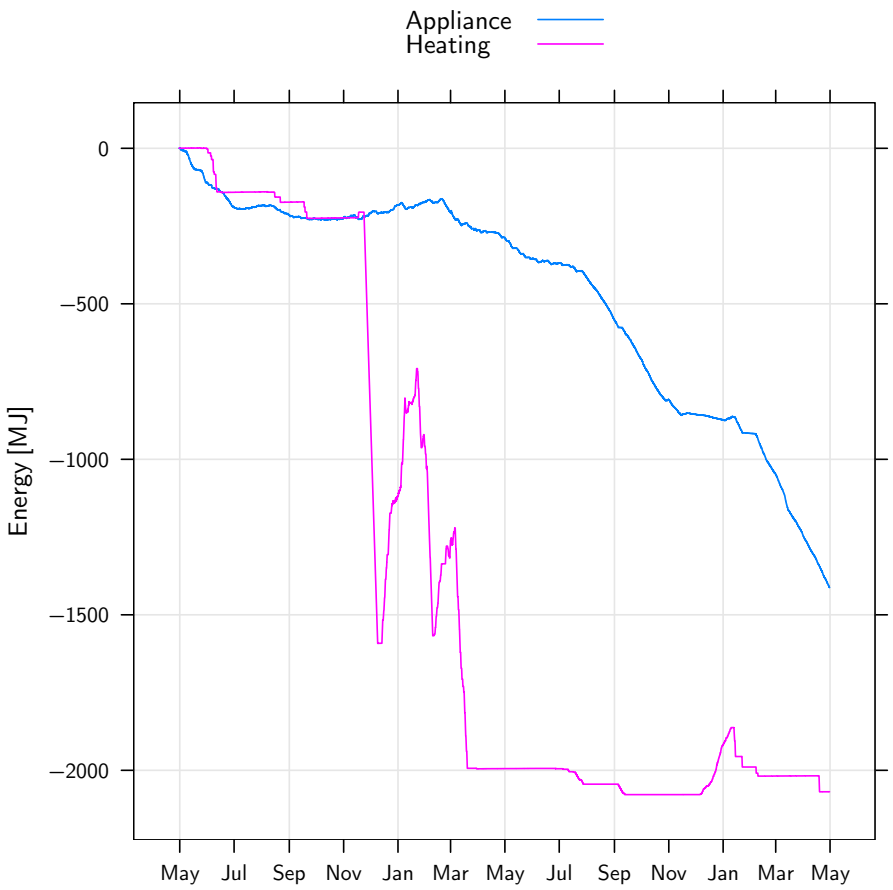


Figure 7.14: Cumulative demand difference between offices 201 and 202. Negative values mean office room 202 consumes more energy. The relative uncertainty on the power demand is estimated at 8% for the heating and at 2% for the appliances.

We do not know for sure why the heating demand of office room 202 is lower than that of office room 201 during the 2006–2007 winter, but perhaps the increase in appliance consumption—such as the PC being left on for days—contributes to enough internal heat gains to void the need for additional heating.

Finally, notice the lack of data between approximately 22 January and 6 February 2007, a period when EIBSERVER lost its EIB connection and stopped recording data.

Figure 7.14 shows the cumulative difference between both offices power demands. Office room 202's heating demand has been almost 2000 MJ higher than that of office room 201, but most of it is due to the unexplained demand in early December. Office room 202 has also the overall higher appliance demand, but between November 2005 and March 2006 that demand was lower than that of office room 201. In 2006, office room 202 consumed about 750 MJ more energy for lighting and office room equipment than office room 201.

Office room 201 has a full-time occupant, whereas office room 202 is used on an irregular basis by one and occasionally two persons. The lighting equipment of office room 201 is an indirect luminaire (144 W installed power) whereas office room 202 is fitted with three ceiling-

mounted fluorescent lights (110 W installed power). The user of office room 201 works on his laptop, while those of office room 202 work with desktops. The occupant of office room 201 sets his thermostat to the lowest possible setpoint (16 °C), whereas the occupants of office room 202 choose intermediate settings (usually 20 °C).

These factors explain why the power demands of both office rooms are so different, although they are fitted with the same shading devices. It will therefore not be possible to make a meaningful comparison between these office rooms.

Figure 7.15 shows the distribution of office room 201's lighting and appliance power demand according to the day of the week and the hour of the day. The patterns for the week's working days are similar: background demand of about 10 W between midnight and 9 a.m.; the laptop or the lighting is switched on between 9 and 10, and most of the power demand happens between 10 and 18. There is often late-night activity on Tuesday evenings, almost no demand on Saturdays and some regular demand on Sundays between 14 and 17. The extremely low (below 0.1 W) demands occured between 14 and 16 January 2006 and are artefacts caused by an interruption of the EIBSERVER program.

It is almost possible to determine the individual power consumption of each appliance in office room 201. Figure 7.16 shows the spectral density of office room 201's power demand. Each peak corresponds likely to the maximum power demand of one electrical appliance, or to a combination of these appliances—but this is out of the scope of this project.

We now examine how the electrical consumption in office room 201 changed between 2006 and 2007, between February and May of these years. The controller was working in 2007 but not in 2006. We will, however, not attempt the comparison of the heating consumption. The heating consumption in 2007 was zero by the end of January, but this is almost certainly due to the much milder winter we had that year and cannot be attributed to the controller alone.

Figure 7.17 shows the cumulative demand difference on the lighting and appliance meter between 2006 and 2007, for the months of February to May. This plot suggests that energy savings of about 12 MJ were achieved in three months compared with the previous year, or 6% savings if one assumes a normal constant yearly consumption of 800 MJ. We caution, however, against drawing any conclusions from this short, single-office study—room 201 was occupied between February and May 2006 for 390.60 hours, but only 321.34 hours in 2007. It is, therefore, impossible to credit our controller alone with these energy savings. They could just as well be due to the shorter total occupancy in 2007 compared to 2006, or even to climate differences. Clearly, more extensive field studies would be necessary to validate the energy savings that can be achieved on real office rooms with our controller.

7.3.2 Simulation runs

When assessing the energy performance of our controller on virtual office rooms, we shall always compare it with the energy performance of the same virtual room running in manual mode (as defined in section 3.2.2). The manual mode will be the baseline against which energy savings and visual discomfort improvements will be evaluated.

We do not expect the controller's performance to be immediately satisfactory upon commissioning. Indeed, as we have seen, its daylighting model needs to acquire training data during weekends. But during the early months of operation, as we have also seen, the data is not always sufficient to guarantee its performance. We cannot, therefore, be sure that the energy savings during the first year will be comparable to those of following years.

We are interested in the long term energy savings possible with an integrated daylighting

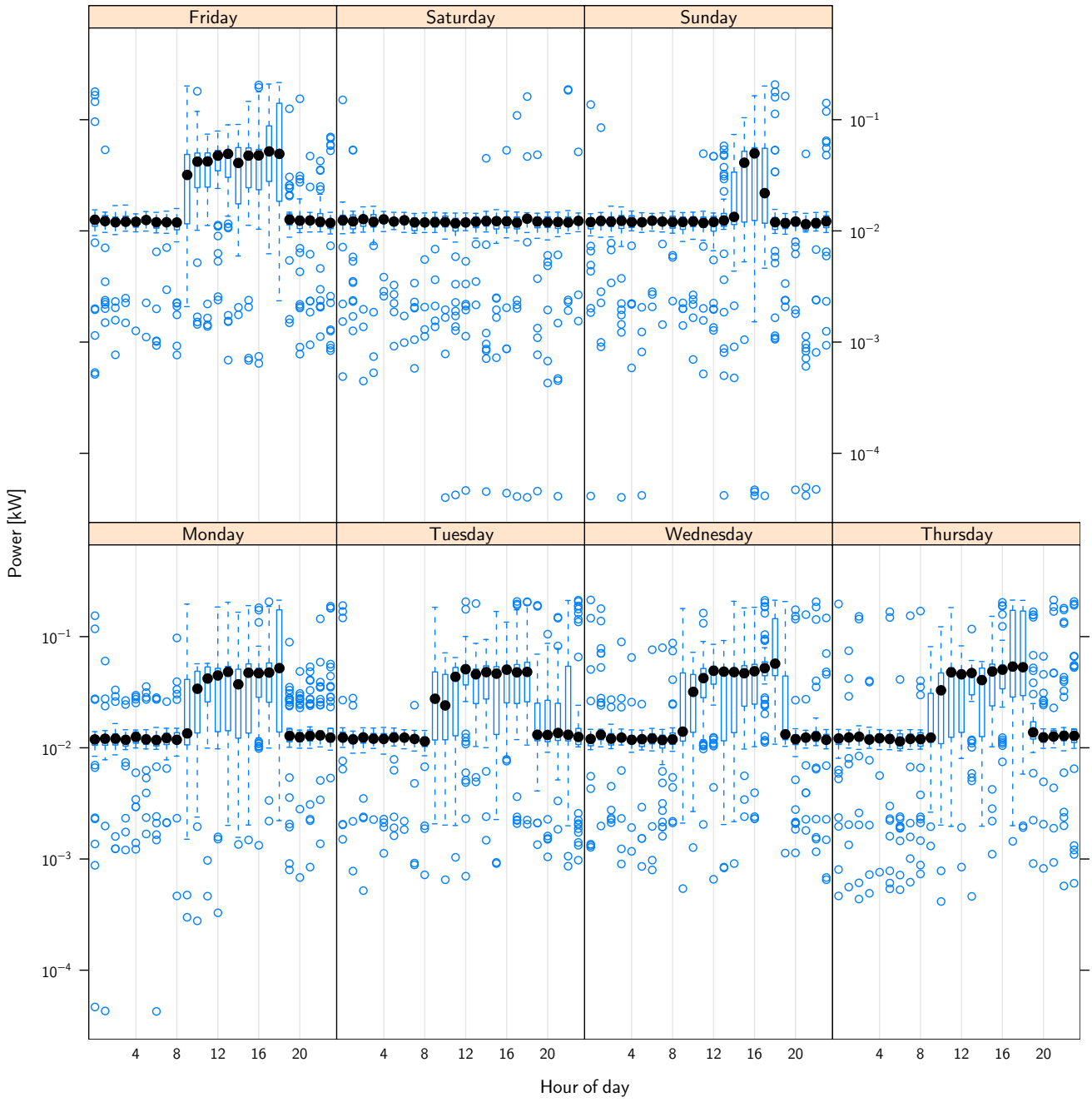


Figure 7.15: Lighting and appliance power pattern in office room 201. 50 points have been randomly sampled for each boxplot.

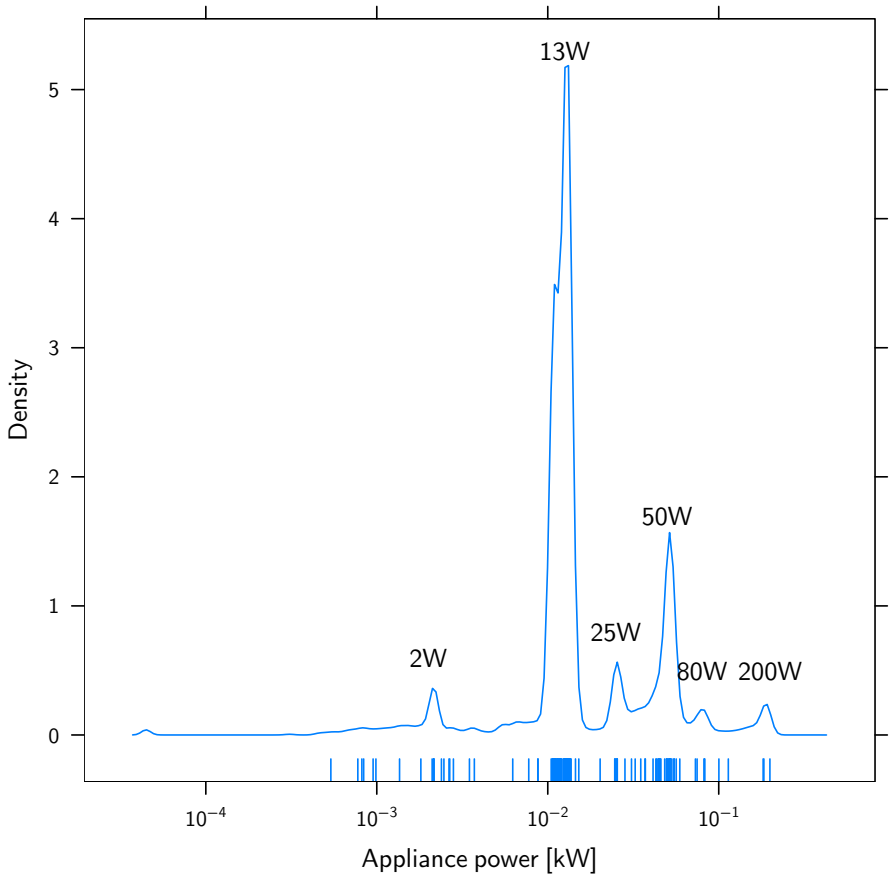


Figure 7.16: Office room 201 lighting and appliance power demand density estimate. 200 randomly chosen data points are shown as tick marks.

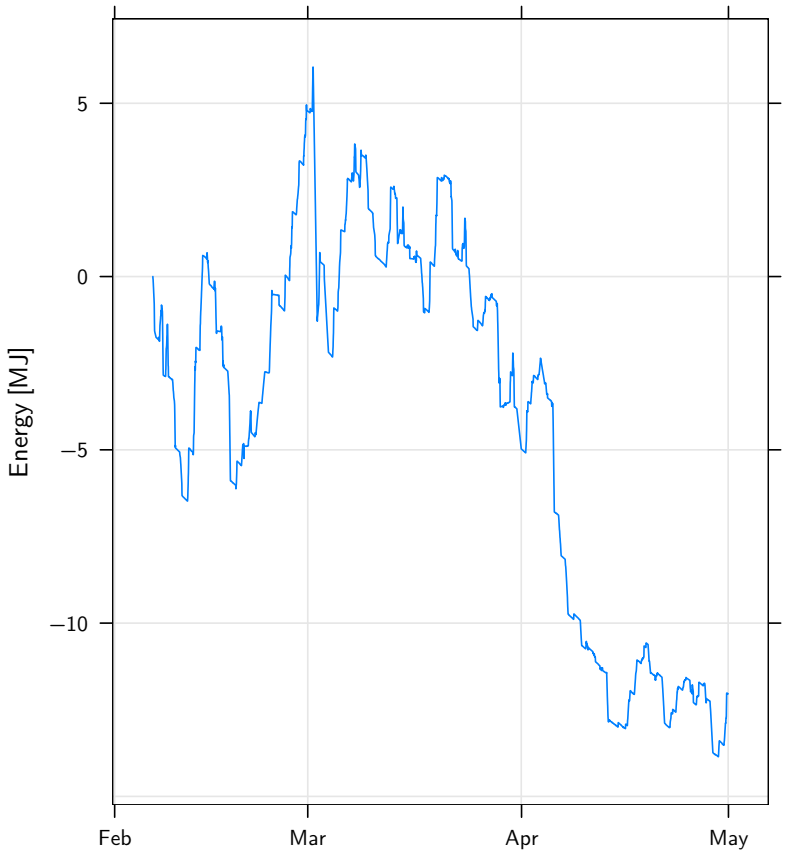


Figure 7.17: Lighting and appliance demand difference between 2006 and 2007 in office room 201, months of February to May. The relative uncertainty is estimated at 2%.

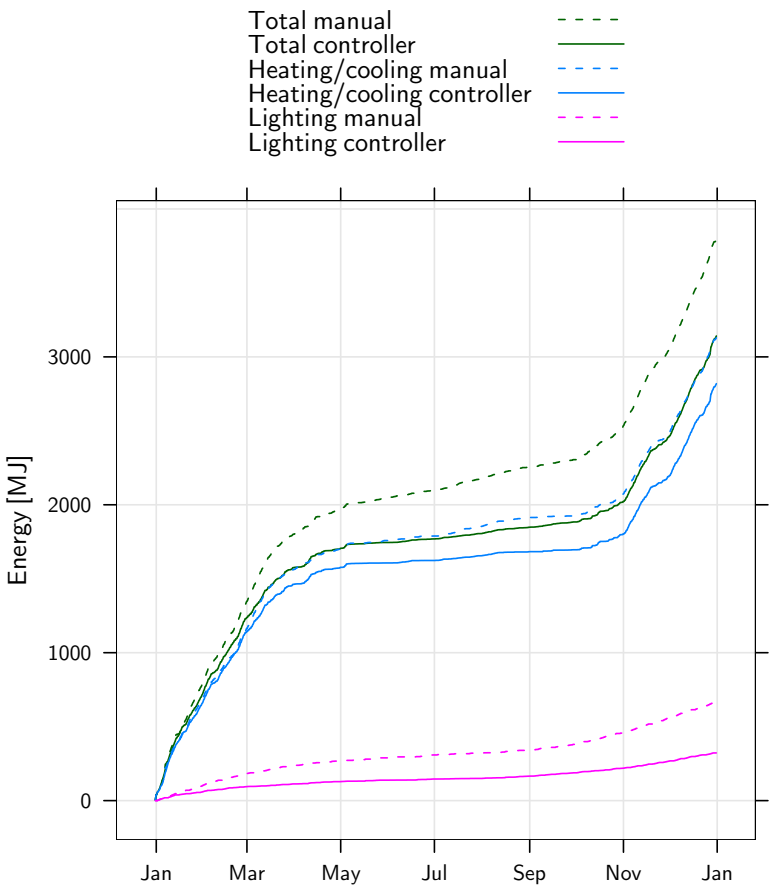


Figure 7.18: Energy demands for the first year of controller operation.

and electric lighting controller. Therefore, unless otherwise noted, all simulation runs will begin with initial daylighting training data corresponding to one year of data acquired during weekends.

One run, however, was started with no initial daylighting data (run `2007-04-12T0900`), to see how the controller behaves during its first year of operation. The energy demand for that office room with and without our controller are shown in Figure 7.18. The controller did surprisingly well, even without initial daylighting data. The energy savings achieved during the first year are comparable to those achieved with initial training data, as we will shortly see, suggesting that the controller adapts itself to the room's daylighting response in not more than a matter of weeks.

The total energy performance of the virtual office rooms in manual or automatic mode after the first year are given in Figure 7.19. The differences in energy demands between manual and controller mode are given in Figures 7.20 and 7.21. These plots show that the total yearly electrical consumption is always reduced in automatic mode, but the energy savings differ according to office room orientation and location.

The total energy savings are smallest when the office room is oriented north, and are almost zero in Brussels. Indeed, the energy demand is even higher for the first part of the year in

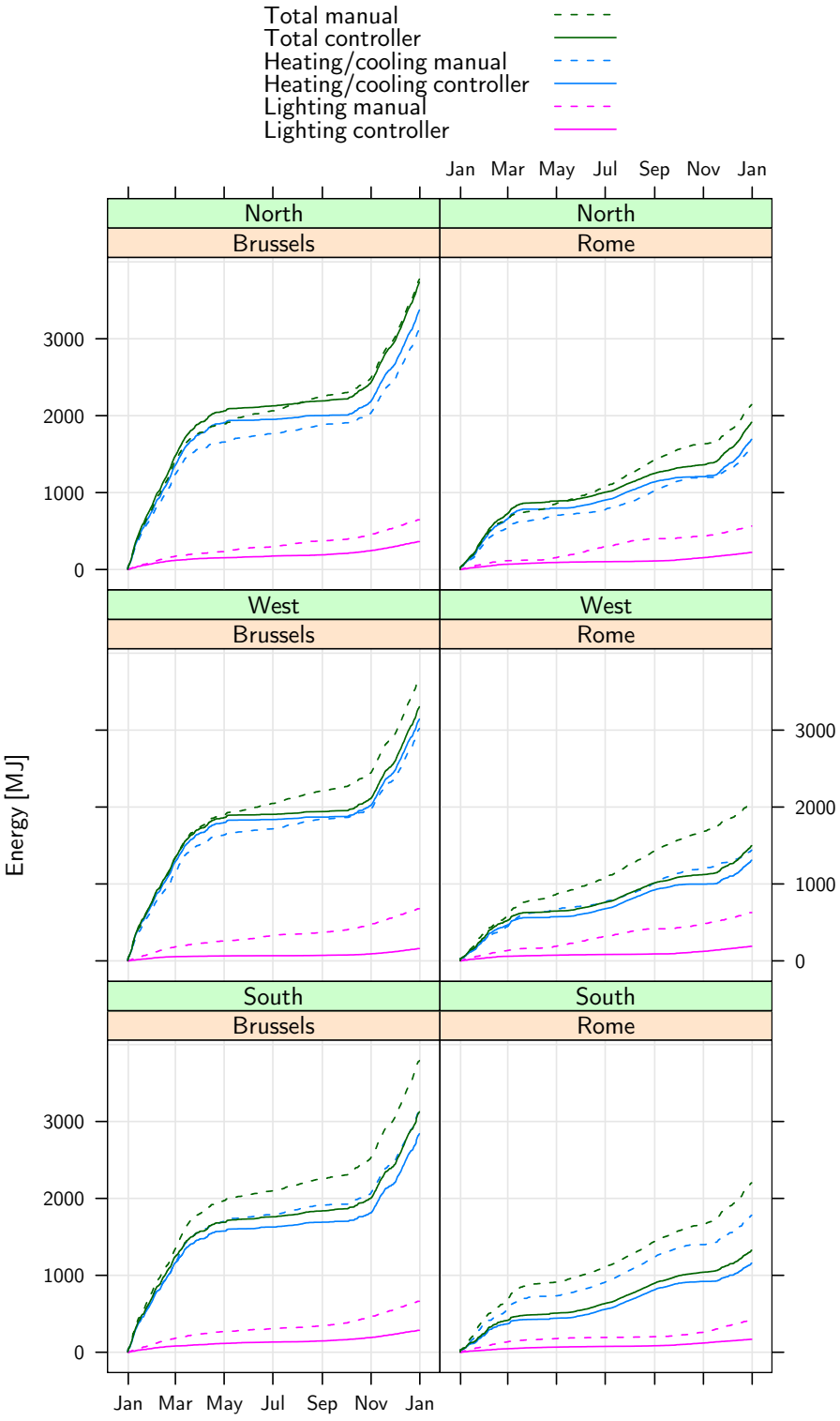


Figure 7.19: Energy demand over time for the virtual office room in manual and controller mode.

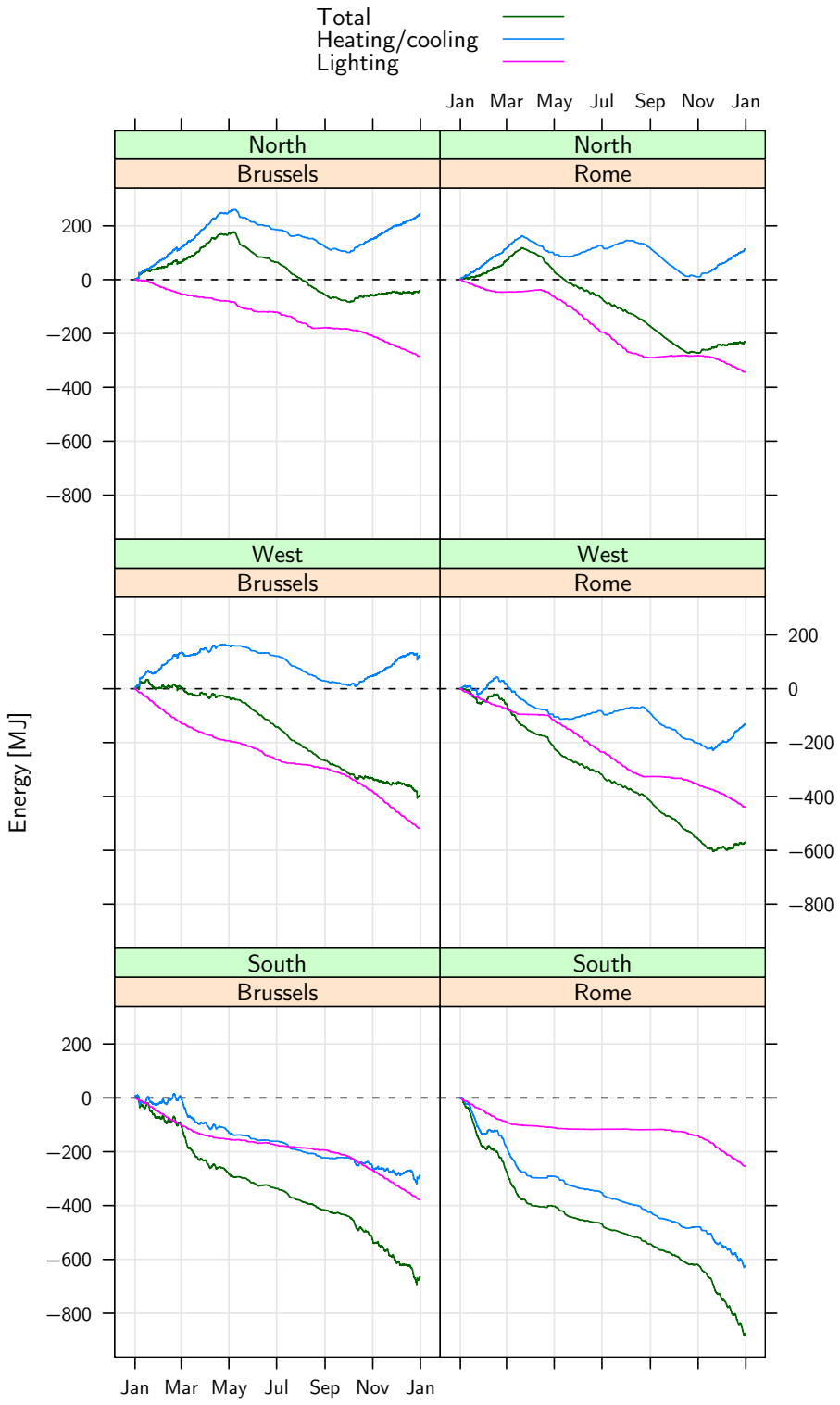


Figure 7.20: Energy savings in controller mode relative to manual mode for the virtual office room.

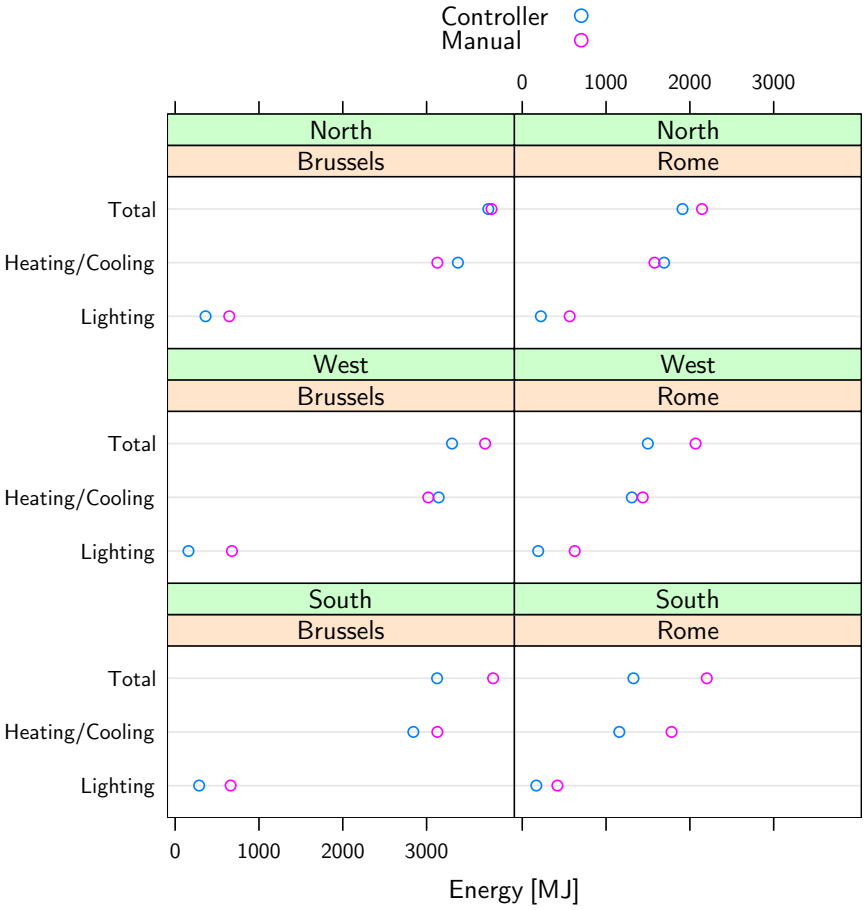


Figure 7.21: Yearly total, heating/cooling and lighting energy demands in manual and controller modes.

Brussels and Rome when the office room is oriented north. We believe this is because the controller cannot use enough solar gains in the absence of the user, hence the higher heating costs during winter.

The heating/cooling costs are slightly higher in controller mode than in manual for an office room oriented north and for the west-oriented Brussels office room. They are only marginally lower for the west-oriented Rome office room. Again, the controller does not have in these cases enough solar gains in this orientation to substantially reduce these heating costs.

On the other hand, the heating/cooling costs are reduced for south-oriented offices—by 9.21% in Brussels and by 35.01% in Rome for the overall year. The savings are much more pronounced in Rome because of the much higher potential solar gains.

In every case, having an automatic controller reduces the electric lighting energy demand. The mean and standard deviation of the relative savings on lighting are $61.3 \pm 11.2\%$. As we have seen in section 7.2.2, these energy savings do not compromise the visual comfort and even accompany an improvement of it. The relative energy savings do not depend significantly on the office room's location nor on its orientation.

We give in Table 7.3 the final yearly energy demands for all combinations of office room orientation, location and mode of operation, for the lighting, heating/cooling and total electrical consumptions. Tables 7.4, 7.5 and 7.6 give the analysis of variance tables for the total, heating/cooling and lighting yearly energy demands as functions of orientation, location and mode. These tables show that the controller has a significant impact on the total and lighting energy demands, whereas the office room location has a significant impact on the total and heating/cooling energy demands. The office room orientation has no statistically significant influence on any energy demand.

The values reported in Table 7.3 suggest that the energy savings achieved with our controller depend on the office room's location and orientation. In statistical terms, the mode, orientation and location factors *interact* with each others. On a large enough data set it is in principle possible to detect such interactions with an analysis of variance, but not when—as in our case—we have only one observation per factor combination. We leave, therefore, this question open to further research.

7.4 Chapter summary

We have tested our controller on-site on a real, occupied office room of the LESO building and monitored that office room's energy consumption. Preliminary results suggest that effective energy savings were indeed achieved on that office room's appliance consumption. We have also progressively trained our controller on that office room's user's preferences and shown that the adaptation is effective. The number of user interactions between 2006 and 2007, controlling for the user's total occupancy during the monitored period, has been almost reduced in half,

Those user preferences have been used to control a virtual office room in two geographical locations and three different orientations. We have shown that the visual comfort improves when running with our controller, reducing the average yearly discomfort probability from 0.44 in manual mode to 0.33 in controller mode.

The savings on lighting energy are comparable for all virtual office rooms. Their mean and standard deviation are $61.3 \pm 11.2\%$. We have seen that this is likely thanks to a more efficient use of the electric lighting at lower power by the controller.

The heating/cooling energy demand of those office rooms that are not oriented south,

Mode	Location	Orientation	Total [MJ]	Savings [%]	Heating/cooling [MJ]	Savings [%]	Lighting [MJ]	Savings [%]
Manual	Brussels	South	3792	N/A	3130	N/A	662	N/A
Manual	Brussels	West	3698	N/A	3021	N/A	678	N/A
Manual	Brussels	North	3775	N/A	3129	N/A	646	N/A
Manual	Rome	South	2202	N/A	1781	N/A	421	N/A
Manual	Rome	West	2067	N/A	1439	N/A	628	N/A
Manual	Rome	North	2144	N/A	1579	N/A	565	N/A
Controller	Brussels	South	3126	17.56	2841	9.23	285	56.95
Controller	Brussels	West	3303	10.68	3143	−4.04	160	76.40
Controller	Brussels	North	3735	1.06	3373	−7.80	362	43.96
Controller	Rome	South	1326	39.78	1158	34.98	168	60.10
Controller	Rome	West	1497	27.58	1308	9.10	189	69.90
Controller	Rome	North	1914	10.73	1692	−7.16	222	60.71

Table 7.3: Yearly energy demands for the virtual office rooms. Negative savings indicate the controller caused higher energy demands.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Orientation	2	14642.46	7321.23	2.87	0.1228
Location	1	679439.66	679439.66	266.52	0.0000
Mode	1	49592.48	49592.48	19.45	0.0031
Residuals	7	17844.88	2549.27		

Table 7.4: Total electrical demand analysis of variance table.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Orientation	2	9566.52	4783.26	1.55	0.2777
Location	1	602421.44	602421.44	194.86	0.0000
Mode	1	2037.71	2037.71	0.66	0.4436
Residuals	7	21640.58	3091.51		

Table 7.5: Heating/cooling electrical demand analysis of variance table.

however, remained either unchanged or slightly higher, but never annulled the savings on lighting energy. The strongest energy savings on heating/cooling were always achieved for south-facing offices, thanks to a more efficient use of free solar gains: 9% in Brussels and 35% in Rome.

The total savings on energy are comparable for south- and west-oriented office rooms, reaching 10–17% in Brussels and 27–40% in Rome, most of which is attributable to savings on heating/cooling energy. The total savings for north-oriented offices are much less satisfactory—but then again, north-oriented offices are not usually plagued by visual discomfort problems, and are thus less likely to need an automatic blinds and electric lighting controller.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Orientation	2	646.26	323.13	0.76	0.5040
Location	1	2315.84	2315.84	5.42	0.0527
Mode	1	31524.96	31524.96	73.83	0.0001
Residuals	7	2988.79	426.97		

Table 7.6: Lighting electrical demand analysis of variance table.

8 Concluding remarks and recommended follow-up

Une multitude de sages est le salut du monde

(Sagesse 6:24)

Political will is a renewable resource.

(Al Gore)

In this work we have described the implementation of a venetian blinds and electric lighting control algorithm in the Java programming language. That prototype runs on its dedicated control PC and communicates to the building's management system through a helper program, running itself on its own dedicated PC.

We have seen how our controller has learned the individual preferences with respect to visual comfort of the occupant in the controlled office room. We have also shown through computer simulations that a controller running according to those individual preferences achieves effective energy savings on office rooms, provided direct solar radiation is available to complement or replace backup heating energy. The visual comfort is, in all cases, improved compared to a non-adaptive controller running according to state-of-the-art principles. The most important numerical results achieved on the computer simulations are summarized in Table 8.1.

In this chapter we suggest some steps that will in our opinion make these energy savings a reality.

8.1 Comprehensive field tests

In spite of the numerous computer simulations that have been carried out in this work, it is evident that more extensive field tests would have been desirable. Indeed, the project's original scope called for prototype installations in Sønderborg, Denmark; in Freiburg im Breisgau, Germany; and in Lyon, France. Time pressure prevented us from realizing these installations, although much infrastructure is already in place. It would not require much work to have copies of our controller running at these locations.

Neither do we know how the algorithm presented in this work would behave in an office room occupied by more than one person. Will the actions of one person conflict with those of the other, or will they complement themselves? This is an important question that must be answered before this system can be implemented commercially.

It is still too early to claim our controller can achieve visual comfort and energy savings in general conditions. We have, in this work, showed that it achieved visual comfort for one office room and energy savings in six different virtual office rooms. Our controller works satisfactorily for $N = 7$ different conditions, which is an encouragingly good start. After all, Euclides himself, who did not know induction, never proved his theorems up to $N = \infty$ but only up to $N = 3$ (Knuth, 1999, p. 336).

Mode	Location	Orientation	Total [MJ]	H/C [MJ]	Light [MJ]	Est. disc. [0–1]
Manual	Brussels	South	3792	3130	662	0.432 ± 0.354
Manual	Brussels	West	3698	3021	678	0.444 ± 0.361
Manual	Brussels	North	3775	3129	646	0.366 ± 0.311
Manual	Rome	South	2202	1781	421	0.513 ± 0.361
Manual	Rome	West	2067	1439	628	0.528 ± 0.398
Manual	Rome	North	2144	1579	565	0.375 ± 0.326
Controller	Brussels	South	3126 (−18%)	2841 (−9%)	285 (−57%)	0.367 ± 0.292
Controller	Brussels	West	3303 (−11%)	3143 (+4%)	160 (−76%)	0.323 ± 0.215
Controller	Brussels	North	3735 (−1%)	3373 (+8%)	362 (−44%)	0.262 ± 0.167
Controller	Rome	South	1326 (−40%)	1158 (−35%)	168 (−60%)	0.367 ± 0.280
Controller	Rome	West	1497 (−28%)	1308 (−9%)	189 (−70%)	0.421 ± 0.315
Controller	Rome	North	1914 (−11%)	1692 (+7%)	222 (−61%)	0.258 ± 0.129

Table 8.1: Yearly total, heating/cooling, and lighting energy demands in all simulated cases. The number in parenthesis are the energy savings achieved by our controller (positive numbers meaning our controller causes more energy demand). The last column gives the average estimated visual discomfort probability. See also Figures 7.9, 7.10, 7.19, 7.20 and 7.21.

8.2 Further improvement towards a commercial controller

Our software was written in Java, a programming language already used in industrial embedded controllers. What steps remain to be taken until it our controller can run on such hardware?

We believe there are three areas where the current implementation must still be refined before it can run on a "real" embedded controller.

First, some elements of the current controller have beeen implemented in the R language. In particular, the density estimator uses the R taut-string algorithm implementation. Such elements must, obviously, be translated in Java, or even implemented as native code for efficiency and speed.

Second, the memory usage could still be improved. At the time of writing, the controller running on LESO office room 201 uses about 100 Mb of memory, which is far too much for an embedded controller. But the current implementation of the daylighting model is very wasteful of memory space (all DAQ measurements are held in memory—144 860 records at the time of writing), and could be much improved.

In particular, the α and β daylighting parameters in Equation 4.15 are determined through a least-squares solution, requiring the controller to keep in memory all the daylighting data. But the *recurrent least-squares* method (Longchamp, 2006) allows one to update the values of α and β as new data becomes available, without requiring old data to be kept. We have not explored this method in this work, but believe it could help keep the controller's memory requirements low.

Third, the components that make up our controller—the daylighting model and the user visual discomfort model, in particular—should be made available as separate software modules. Instead of releasing the whole controller as a single `eccobuild.jar` executable file, it would be wiser to release it as a series of smaller jarfiles, each providing a well-defined service to the others. The current model for such an architecture of small, independent Java modules

that provide services to each others is the OSGi[1] platform definition of *bundles*—an architecture accepted by the Java Community Process Executive Commitee as a Java standard[2] only days before this manuscript went to press. Our controller should, therefore, be split into constituent OSGi bundles, each of which could then be made available to the industry.

8.3 Latent variables discovery

In this work we have modeled the user's visual discomfort probability as a node in a very simplified bayesian network. There is no reason why more nodes could not be added to this network, providing a better understanding of the comfort the building provides.

In an utopic situation, a computer program could automatically detect the sensors and actuators in the building, and build up by itself the corresponding bayesian network by looking for relationships in the monitored data. But this kind of system remains, for the moment, a dream. A system that discovers the structure of a bayesian network from monitored data is difficult to build and is an active topic of research.

We do propose, however, to research the existence and discoverability of *latent* variables in bayesian networks that describe buildings. A latent variable (see e.g. Bartholomew and Knott, 1999) is a variable that is not directly measurable but that is causally connected to other variables. For example, a room occupancy is a latent variable causally connected to the occupancy sensor output. This distinction is important, because occupancy sensors can malfunction or mistakenly detect a user as absent when they are sitting very still.

Dodier et al. (2006) have built a bayesian network between several occupancy sensors in the same room and the "real" manually recorded occupancy. They show how this bayesian network can be used to improve the accuracy in occupancy detection, even though the accuracy of the sensors is not perfect.

But the weakness in their approach is the reliance on the manual recording of the real occupancy, necessary to build the relationships in the bayesian network. Instead, we propose to research the possibility to detect the state of such latent variables automatically, without human intervention.

Besides the room's real occupancy, most aspects of environmental comfort could be modeled as latent variables. The thermal comfort, the visual comfort and the air quality are all variables that are causally connected to the user's behaviour, and that could be discovered in this manner. There is in principle no limit to the number of nodes that could be added to our humble initial sketch—each one improving the acceptance by building users of automatic control systems.

[1] `http://www.osgi.org`

[2] Java Specification Request 291: Dynamic Component Support for Java SE, `http://jcp.org/en/jsr/detail?id=291`

A Hardware independent building control API

In this appendix we develop further the idea of a standardized interface for building management systems. As we have seen in chapter 6, the EIBSERVER program exposes such a remote interface through the Java RMI protocol for the control of building sensors and actuators. We will also discuss how such an interface can be made to work independently of the underlying physical hardware.

It is difficult for smart building control algorithms developed by the academia to find their way in commercial BMS. Building physicists often develop highly advanced, self-commissioning, user-adaptive, energy-saving systems that unfortunately operate only on hardware specifically designed and built for the purpose of their projects. Few control systems developers master the programming skills necessary to "port" their algorithms to commercial systems. Even when commercial hardware is used, it is difficult to field-test the systems on building installations that do not use exactly the same hardware.

This often leads to much code rewriting and great costs in terms of time spent managing the extra complexity. We propose here a "middleware" program between the higher-level development of smart control algorithms and the lower-level physical hardware. Depending on the specific BMS being used, the program either discovers by itself the available actuators and sensors or reads in their definitions from a configuration file. The corresponding set of objects is then made available to the algorithm developer. The main advantage for the latter is that the algorithm needs have no knowledge of what kind of BMS is used, whether it is a home-grown one or a commercial system—indeed, it does not even need to know whether a building exists at all, since the control program can be plugged into a building simulator exposing an identical interface without seeing any difference.

The algorithm implementer is thus presented with a uniform set of interfaces representing the different controllable actuators in the building and the available sensors. For instance, the system provides a well-defined and well-documented `Blind` interface with obviously required methods such as `Blind::getPosition()` or `Blind::setPosition(double)`. Sensor interfaces are also defined, such as for instance `Sun`, with methods such as `Sun::globalIrradiation()`.

The price to pay for this simplicity is the effort required to write a very limited number of drivers for elementary operations on the BMS—essentially, programs that know how to read/write boolean/analog values from/to the BMS. Smart building control algorithm developers would benefit the most from this middleware program as it will help them concentrate on their field of expertise, namely the specification of the behaviour of building elements as a function of available sensors and of their values. It keeps their minds free from worries about the specific implementation of the control system, which needs not be decided before the design of the control system is finished. Commercial BMS experts should implement or help implement the necessary drivers specific to their systems until drivers are available for the major commercial BMS in use today.

The simplicity of the program is achieved by two means. First, by the application of object-oriented design principles to the problem of building control, and the identification of

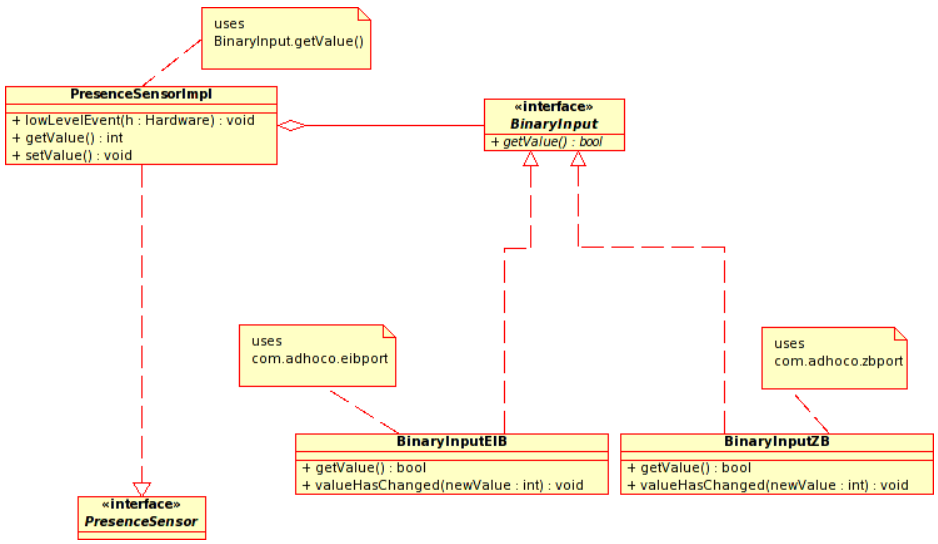


Figure A.1: Implementation example of the Bridge design pattern in order to have an occupancy sensor that can at run-time be attached to an EIB system or a ZigBee system. The control algorithm writer never sees these implementation details and has only access to the `PresenceSensor` interface. The `BinaryInputEIB` and `BinaryInputZB` drivers use libraries developed by Adhoco AG to communicate with the respective BMS.

elementary classes. From this analysis it emerged that a device such as a blind actuator can be modeled either as a blind (the algorithm's point of view), or as an object that sends and receives analog values on a BMS (the BMS's point of view).

Second, by the application of the design pattern known in the object-oriented community as a "Bridge" (Gamma et al., 1995), separating the implementation of a class representing, for instance, a presence sensor from the abstract class responsible for reading a boolean value from the BMS. The concrete classes inheriting from the latter abstract class are what we call the drivers for the different BMS. An example is given in Figure A.1 for a presence sensor that will at run-time be attached to an EIB sensor or to a sensor communicating with the ZigBee[1] standard, the low-bandwidth wireless communication and control standard. A presence sensor and an alarm system thus both use the same class responsible for reading boolean values from the system, promoting code reuse and keeping the program simple, but both remain distinct objects from the algorithm's point of view.

This program has been developed in Java in a joint collaboration between LESO-PB and the Winterthur-based Adhoco AG[2] firm. It is currently being used as middleware for algorithms developed by LESO-PB, and is also used by Adhoco AG for the development of their own proprietary smart building control algorithms.

[1] `http://www.zigbee.org`
[2] `http://www.adhoco.com`

B Statistical methods

In this appendix we give a succint overview of the statistical methods used in this work. In no way is this appendix a supplement for a statistics textbook—the interested reader is urged to consult such a work for a more rigorous derivation of these results. Morgenthaler (2001) is a very good introductory text for engineers and researchers. Davison (2003) and Venables and Ripley (2002) are more advanced texts, the latter being particularly geared towards R/S-PLUS users.

We illustrate this appendix with example code in R, where appropriate.

B.1 Analysis of variance in R

Physical experiments usually consist in repeating a measurement, each time varying one or a combination of variables. In a *designed* experiment, some of these variables will take on values from a finite set of levels designed by the experimenter, usually termed *factors*.

Consider the following example, taken from Cleveland et al. (1992). We examine the effect on the yield of some process from two different catalysts `A` and `B`, at two different concentrations and two different temperatures.

```
> catalyst <- data.frame(Temp = as.factor(rep(c(160, 180), 4)),
+     Conc = as.factor(rep(c(20, 40), 2, each = 2)), Cat = as.factor(rep(c("A",
+         "B"), each = 4)), Yield = c(60, 72, 54, 68, 52, 83, 45,
+         80))
> catalyst

  Temp Conc Cat Yield
1  160   20   A    60
2  180   20   A    72
3  160   40   A    54
4  180   40   A    68
5  160   20   B    52
6  180   20   B    83
7  160   40   B    45
8  180   40   B    80
```

An analysis of variance is the preferred tool to understand which factors influence the outcome of the experiment, and whether their influence is statistically significant.

The outcome of such an experiment is often modeled as an additive model in all the factors. R has a rich syntax for describing statistical models, and the syntax for modeling `Yield` as an additive model in the other factors is `Yield ~ Temp + Conc + Cat`. This can also be expressed as the shorthand `Yield ~ .` where the dot refers to all the other factors in the data.

Running an analysis of variance in R yields the best-fitting coefficients of each factor:

```
> cat.aov <- aov(Yield ~ ., catalyst)
> coef(cat.aov)

(Intercept)     Temp180      Conc40        CatB
       54.5        23.0        -5.0         1.5
```

For the temperature factor, for instance, this tells us that increasing the temperature from 160 °C to 180 °C increases the yield on average by 23.0.

R also provides a summary output for analysis of variance objects:

```
> summary(cat.aov)

            Df  Sum Sq Mean Sq F value  Pr(>F)
Temp         1 1058.00 1058.00 20.6439 0.01047 *
Conc         1   50.00   50.00  0.9756 0.37920
Cat          1    4.50    4.50  0.0878 0.78173
Residuals    4  205.00   51.25
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This summary output tells us which factors contribute significantly to the linear model, and the number of stars after each row indicates the level of statistical significance. The important value to the experimenter is the `F value` column. If the factor does not contribute to the linear model, the value in that column should be drawn from an F distribution with parameters $p - q$ and $n - p - 1$, where p is the number of factors, q is the degrees of freedom associated with that factor (given in the first column), and n is the number of experiments.

The `Pr(>F)` column gives the probability that an F distribution with these parameters should yield a value at least as large as that given in the `F value` column. A factor is said to be statistically significant if this probability is lower than a given threshold, usually 5%.

B.2 Linear models

R provides the `lm` command for fitting linear models. We illustrate it with a simple linear $y = 20 \times x$ model, with random noise added to the y variable. Let's generate this fake data:

```
> x <- 0:20
> y <- 20 * x + rnorm(21, sd = 20)
> dummy <- data.frame(x = x, y = y)
```

Here the variable `x` is a vector of integers between 0 and 20, and `y` holds `20*x` plus random noise.

The best-fitting linear model is shown in Figure B.1. The parameters of the fit are obtained with this command:

```
> dummy.fit <- lm(y ~ x, dummy)
> summary(dummy.fit)
```

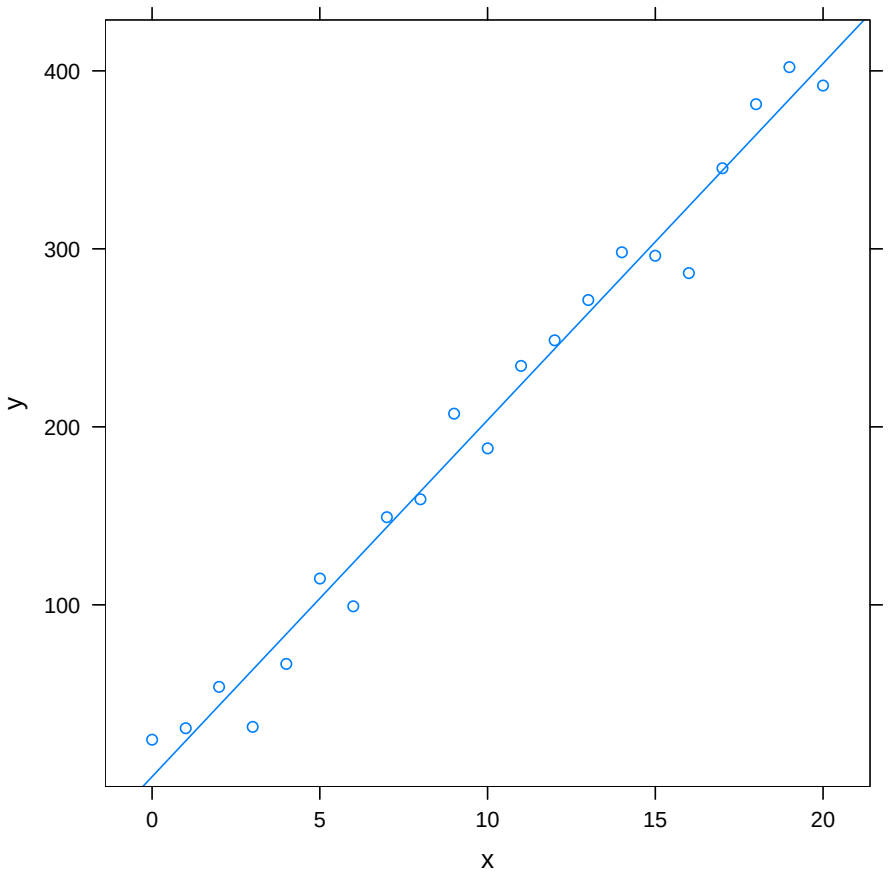


Figure B.1: Best-fitting linear model for dummy data.

```
Call:
lm(formula = y ~ x, data = dummy)

Residuals:
   Min     1Q Median     3Q    Max
-37.60 -12.41   5.52  11.19  23.59

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.4709     7.5648   0.459    0.652
x            20.0323     0.6471  30.958   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.96 on 19 degrees of freedom
Multiple R-Squared: 0.9806,        Adjusted R-squared: 0.9795
F-statistic: 958.4 on 1 and 19 DF,  p-value: < 2.2e-16
```

From this output we have the estimates of the intercept and the slope of the linear model, and the errors on these estimates.

The `t value` field is the ratio between the estimate and the error, and this ratio should be compared to a Student distribution. The last field, `Pr(>|t|)`, gives the probability that a sample from this Student distribution should give a value at least as large as the one given in the `t value` field. A coefficient is said to be significantly different from zero if this probability is lower than a threshold, usually 5%.

C Source code

C.1 Companion website

All the material, including original data, that has been used in this work can be found on its companion website:

`http://visnet.ch/smartbuildings/bayesian-controller/`

The `eccobuild.tar.gz` file contains all the source code of the controller itself, mostly written in Java but with some elements in R. It can be built by following the instructions given in the `README` file.

The `thesis.tar.gz` file contains all the sources and data files used in the production of this manuscript. All figures and a PDF of the thesis can be re-built by executing the command `make`.

The `scripts` directory in the thesis distribution contains some scripts programs used in this work. In particular, it contains the `eibmon.pl` Perl script used to convert the text data produced by EIBSERVER into MySQL data.

The files for the SIMBAD could not be made publicly available. SIMBAD's licence does not allow it, and the illuminance data files are too voluminous, even compressed. Contact the author directly for information.

Not all results of the simulation runs could be made available either, but only those directly used in this work (and necessary for generating this manuscript).

C.2 Precalculated illuminance files naming convention

The precalculated illuminance files are given names that, for venetian blinds, will match the following regular expression: `⌜^illum_(v\d{3})_(\d+)_(\d+)_N\.ill$⌟` where the first group refers to an entry in Table C.1. The second group refers to the blinds' position (0 is closed, 90 is 90% open) and the third group refers to the blinds' slat angle (0 is horizontal, 66 is closed, in steps of 6°). For example, file `illum_v001_0_0_N.ill` refers to the south-oriented room in Brussels with completely closed venetian blinds.

C.3 Astronomical Almanac algorithm R implementation

Here we give our translation in R of Michalsky's Fortran implementation of the *Astronomical Almanac* algorithm for calculating the sun's position. Seidelmann (1992) should be consulted for the details of this algorithm.

```
1 sunPosition <- function(year, month, day, hour=12, min=0, sec=0,
2                         lat=46.5, long=6.5) {
3   twopi <- 2 * pi
```

Variant number	Blinds	Window orientation	City
v001	Venetian	South	Brussels
v002	Venetian	West	Brussels
v003	Venetian	North	Brussels
v004	Venetian	South	Rome
v005	Venetian	West	Rome
v006	Venetian	North	Rome
v007	Roller	South	Brussels
v008	Roller	North	Brussels
v009	Roller	West	Brussels
v010	Roller	South	Rome
v011	Roller	North	Rome
v012	Roller	West	Rome

Table C.1: Office variant numbers.

```
4   deg2rad <- pi / 180
5
6   # Get day of the year, e.g. Feb 1 = 32, Mar 1 = 61 on leap years
7   month.days <- c(0,31,28,31,30,31,30,31,31,30,31,30)
8   day <- day + cumsum(month.days)[month]
9   leapdays <- year %% 4 == 0 & (year %% 400 == 0 | year %% 100 != 0) &
10              day >= 60
11  day[leapdays] <- day[leapdays] + 1
12
13  # Get Julian date - 2400000
14  hour <- hour + min / 60 + sec / 3600 # hour plus fraction
15  delta <- year - 1949
16  leap <- trunc(delta / 4) # former leapyears
17  jd <- 32916.5 + delta * 365 + leap + day + hour / 24
18  # The input to the Atronomer's almanach is the difference between
19  # the Julian date and JD 2451545.0 (noon, 1 January 2000)
20  time <- jd - 51545.
21
22  # Ecliptic coordinates
23  # Mean longitude
24  mnlong <- 280.460 + .9856474 * time
25  mnlong <- mnlong %% 360
26  mnlong[mnlong < 0] <- mnlong[mnlong < 0] + 360
27  # Mean anomaly
28  mnanom <- 357.528 + .9856003 * time
29  mnanom <- mnanom %% 360
30  mnanom[mnanom < 0] <- mnanom[mnanom < 0] + 360
31  mnanom <- mnanom * deg2rad
32  # Ecliptic longitude and obliquity of ecliptic
33  eclong <- mnlong + 1.915 * sin(mnanom) + 0.020 * sin(2 * mnanom)
34  eclong <- eclong %% 360
```

```
35    eclong[eclong < 0] <- eclong[eclong < 0] + 360
36    oblqec <- 23.429 - 0.0000004 * time
37    eclong <- eclong * deg2rad
38    oblqec <- oblqec * deg2rad
39
40    # Celestial coordinates
41    # Right ascension and declination
42    num <- cos(oblqec) * sin(eclong)
43    den <- cos(eclong)
44    ra <- atan(num / den)
45    ra[den < 0] <- ra[den < 0] + pi
46    ra[den >= 0 & num < 0] <- ra[den >= 0 & num < 0] + twopi
47    dec <- asin(sin(oblqec) * sin(eclong))
48
49    # Local coordinates
50    # Greenwich mean sidereal time
51    gmst <- 6.697375 + .0657098242 * time + hour
52    gmst <- gmst %% 24
53    gmst[gmst < 0] <- gmst[gmst < 0] + 24.
54
55    # Local mean sidereal time
56    lmst <- gmst + long / 15.
57    lmst <- lmst %% 24.
58    lmst[lmst < 0] <- lmst[lmst < 0] + 24.
59    lmst <- lmst * 15. * deg2rad
60
61    # Hour angle
62    ha <- lmst - ra
63    ha[ha < -pi] <- ha[ha < -pi] + twopi
64    ha[ha > pi] <- ha[ha > pi] - twopi
65
66    # Latitude to radians
67    lat <- lat * deg2rad
68
69    # Azimuth and elevation
70    el <- asin(sin(dec) * sin(lat) + cos(dec) * cos(lat) * cos(ha))
71    az <- asin(-cos(dec) * sin(ha) / cos(el))
72    elc <- asin(sin(dec) / sin(lat))
73    az[el >= elc] <- pi - az[el >= elc]
74    az[el <= elc & ha > 0] <- az[el <= elc & ha > 0] + twopi
75
76    el <- el / deg2rad
77    az <- az / deg2rad
78    lat <- lat / deg2rad
79
80    return(list(elevation=el, azimuth=az))
81  }
```

C.4 Machine synchronization

All PCs used in this project are kept synchronized by placing the following `sync-root` script in their `/etc/cron.hourly/` directory. It will run the `ntpdate` command about once an hour, synchronizing with a Network Time Protocol server in Zürich.

```
#!/bin/sh
#
# $Id: sync-clock,v 1.6 2003/09/23 21:39:29 jmates Exp $
#
# Use ntpdate to get rough clock sync.

NTPDATE=/usr/sbin/ntpdate
SERVER="swisstime.ethz.ch"

# if running from cron (no tty available), sleep a bit to space
# out update requests to avoid slamming a server at a particular time
if ! test -t 0; then
  MYRAND=$RANDOM
  MYRAND=${MYRAND:=$$}

  if [ $MYRAND -gt 9 ]; then
    sleep `echo $MYRAND | sed 's/.*\(..\)$/\1/' | sed 's/^0//'`
  fi
fi

$NTPDATE -su $SERVER

# update hardware clock on Linux (RedHat?) systems
if [ -f /sbin/hwclock ]; then
  /sbin/hwclock --systohc
fi
```

Bibliography

Richard Alley et al. Climate change 2007: the physical science basis. Technical report, International Panel on Climate Change, 2007.

René Altherr and Jean-Bernard Gay. A low environmental impact anidolic facade. *Building and Environment*, 37(12):1409–1419, 2002.

Ion Androutsopoulos et al. An evaluation of naive bayesian anti-spam filtering. In *Proceedings of the workshop on Machine Learning in the New Information Age*, 2000.

Stephan Bader and Heinz Bantle. Das schweizer Klima im Trend. Temperatur- und Niederschlagsentwicklung 1864-2001. *Veröffentlichung der MeteoSchweiz*, 68:45, 2004.

J. J. Balder. Erwünschte Leuchtdichten in Büroräumen. *Lichttechnik*, 9(9), 1957.

David J. Bartholomew and Martin Knott. *Latent Variable Models and Factor Analysis*. Hodder Arnold, 1999.

A. R. Bean and A. G. Hopkins. Task and background lighting. *Lighting Research and Technology*, 12(3):135–139, 1980.

S. H. A. Begemann, G. van den Beld, and J. A. D. Tenner. Daylight, artificial light and people in an office environment, overview of visual and biological responses. *Industrial Ergonomics*, 20:231–239, 1997.

D. M. Berson, F. A. Dunn, and M. Takao. Phototransduction by retinal ganglion cells that set the circadian clock. *Science*, 295:1070–1073, 2002.

Manuel Blanco-Muriel, Diego C. Alarcón-Padilla, Teodoro López-Moratalla, and Martín Lara-Coira. Computing the solar vector. *Solar Energy*, 70(5):431–441, 2001.

H. W. Bodmann. Beleuchtungsniveaus und Sehtätigkeit. *Internationale Licht Rundshau*, 13 (2):41–47, 1962.

H. W. Bodmann, G. Söllner, and E. Voit. Bewertung der Beleuchtungsniveaus bei verschiedenen Lichtarten. *Compte Rendu CIE, Vienna*, pages 502–509, 1963.

Adrian W. Bowman and Adelchi Azzalini. *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford Science Publications, 1997.

R. Boyce. The influence of illumination levels on prolonged work performance. Technical report, The Electricity Concil Research Centre, Capenhurst/England, 1968.

Laird Breyer. dbacl – a digramis bayesian classifier, 2004. URL `http://dbacl.sourceforge.net`. Last checked 1 November 2006.

P. Chauvel, J. B. Collins, R. Dogniaux, and J. Longmore. Glare from windows: current views of the problem. *Lighting Research and Technology*, 14(1):31–46, 1982.

CIE. Discomfort glare in the interior working environment. Technical report, Commission Internationale de l'Eclairage, 1983.

CIE. Guide on interior lighting, second edition. Technical report, Commission Internationale de l'Eclairage, 1986.

CIE. Discomfort glare in interior lighting. Technical report, Commission Internationale de l'Eclairage, 1995.

CIE-1970. Daylight. Technical report, Commission Internationale de l'Eclairage, 1970.

CIE-2001. Lighting of indoor work places. Technical report, ISO 8995:2002/CIE S 008/E:2001, 2001.

CIE-2002. CIE collection on glare 2002. Technical report, Commission Internationale de l'Eclairage 146:2002/147:2002, 2002.

Joe Clarke. *Energy Simulation in Building Design*. Butterworth Heineman, second edition edition, 2001.

W. S. Cleveland, E. Grosse, and W. M. Shyu. *Statistical Models in S*, chapter 8, Local regression models. Wadsworth & Brooks/Cole, 1992.

William S. Cleveland. *Visualizing Data*. Hobart Pr, 1993.

Gilles Courret and Bernard Paule. Gestion optimale des stores à lamelles orientables. Technical report, Université de Genève, 1993.

Ward P. Cunningham and Barbara Saigo. *Environmental Science: a Global Concern*. William C. Brown, 1995.

P. L. Davies and A. Kovac. Local Extremes, Runs, Strings and Multiresolution. *Annals of Statistics*, 29(1):1–65, 2001.

P. L. Davies and A. Kovac. Densities, spectral densities and modality. *Annals of Statistics*, 32:1093–1136, 2004.

A. C. Davison. *Statistical Models*. Cambridge University Press, 2003.

Delta-T Devices Ltd. Sunshine sensor BF3, 2006. URL `http://www.delta-t.co.uk/products.html?product2005092016583`. Last checked 23 october 2006.

Robert H. Dodier, Gregor P. Henze, Dale K. Tiller, and Xin Guo. Building occupancy detection through sensor belief networks. *Energy and Buildings*, 38(9):1033–1043, 2006.

Pedro Domingos and Michael Pazzani. Beyond independence: conditions for the optimality of the simple bayesian classifier. In *Proceedings of the 13th International Conference on Machine Learning*, 1996.

EC-2002. Directive 2002/91/ec of the European Parliament and of the concil of 16 december 2002 on the energy performance of buildings, 2002.

Charles Elkan. Naive bayesian learning. Technical report, Department of Computer Science, Harvard University, 1997.

Anna Marie Nygård Ferguson. *Predictive thermal control of building systems*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 1990.

Dietert Fischer. Optimale Beleuchtungsniveaus in Arbeitsräumen. *Lichttechnik*, 22(2), 1970.

Michelle Foster and Tadj Oreszczyn. Occupant control of passive systems: the use of venetian blinds. *Building and Environment*, 36:149–155, 2001.

Martin Fowler. *UML Distilled*. Addison-Wesley, third edition edition, 2003.

Anca D. Galasiu and Jennifer A. Veitch. Occupant preferences and satisfaction with the luminous environment and control systems in daylit offices: a literature review. *Energy and Buildings*, 38(7):728–742, 2006.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley, 1995.

Harry Goldstein and William Sweet. Joules, BTUs, quads—let's call the whole thing off. *IEEE Spectrum*, January 2007.

Paul Graham. A plan for spam, 2002. URL `http://www.paulgraham.com/spam.html`. Last checked 1 November 2006.

Etienne Grandjean. *Ergonomics in Computerized Offices*. Taylor and Francis, 1987.

Etienne Grandjean. *Fitting the Task to the Man*. Taylor and Francis, 1988.

William Grosso. *Java RMI*. O'Reilly Media, 2001.

Antoine Guillemin. *Using genetic algorithms to take into account user wishes in an advanced building control system*. PhD thesis, LESO-PB/EPFL, 2003.

Antoine Guillemin and Simone Molteni. An energy-efficient controller for shading devices self-adapting to user wishes. *Buildings and Environment*, 37(11):1091–1097, 2002.

Antoine Guillemin and Nicolas Morel. Experimental assessment of three automatic building controllers over a 9-month period. In *Proceedings of the CISBAT 2003 conference*, pages 185–190, Lausanne, Switzerland, 2003.

David Heckerman. A tutorial on learning bayesian networks. Technical report, Microsoft Research, 1995.

IEA-2007. IEA statistics: balances by country/region. URL `http://www.iea.org/Textbase/stats/prodresult.asp?PRODUCT=Balances`. Last checked 11 May 2007.

V. Jacobson. Compressing TCP/IP headers for low-speed serial links. RFC 1144, 1990.

George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *11th Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

Phil Jones and Mike Salmon. Temperature data (`HadCRUT3` and `CRUTEM3`), 2006. URL `http://www.cru.uea.ac.uk/cru/data/temperature/`. Last checked 5 January 2007.

C. D. Keeling and T. P. Whorf. Atmospheric CO_2 records from sites in the SIO air sampling network. In *Trends: A Compendium of Data on Climate Change*. Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, U.S. Department of Energy, Oak Ridge, Tenn., U.S.A., 2005.

David C. Knill and Alexandre Pouget. The bayesian brain: the role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, 27(12):712–719, 2004.

Donald E. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley, 1999.

D. Kolokotsa, G. Saridakis, A. Pouliezos, and G.S. Stavrakakis. Design and installation of an advanced EIB fuzzy indoor comfort controller using Matlab. *Energy and Buildings*, 38: 1084–1092, 2006.

Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, 2003.

A. Kovac. Smooth functions and local extreme values. *Computational Statistics & Data Analysis*, 51(10):5155–5171, 2007.

Thomas S. Kuhn. *The Structure of Scientific Revolutions*. B & T, 1996.

Tilmann E. Kuhn. Solar control: comparison of two new systems with the state of the art on the basis of a new general evaluation method for facades with venetian blinds or other solar control systems. *Energy and Buildings*, 38(6):661–672, 2006a.

Tilmann E. Kuhn. Solar control: a general evaluation method for facades with venetian blinds or other solar control systems. *Energy and Buildings*, 38(6):648–660, 2006b.

Mateja Trobec Lah, Borut Zupančič, Jože Peternelj, and Aleš Krainer. Daylight illuminance control with fuzzy logic. *Solar Energy*, 80:307–321, 2006.

Greg Ward Larson and Rob Shakespeare. *Rendering with Radiance, Revised Edition*. Space & Light, 2003.

E. S. Lee and S. E. Selkowitz. The New York Times Headquarters daylighting mockup: monitored performance of the daylighting control system. *Energy & Buildings*, 38(7):914–929, 2006.

E. S. Lee, D. L. DiBartolomeo, and S. E. Selkowitz. Thermal and daylighting performance of an automated venetian blind and lighting system in a full-scale private office. *Energy and Buildings*, 29:47–63, 1998.

LEED. Leed for new construction version 2.2, 2005. URL `http://www.usgbc.org/LEED`. Last checked 3 June 2007.

M. A. Lehar and L. R. Glicksman. Rapid algorithm for modeling daylight disctributions in office buildings. *Buiding and Environment*, 42:2908–2919, 2007.

David Lindelöf and Nicolas Morel. A field investigation of the intermediate light switching by users. *Energy and Buildings*, 38(7):790–801, 2006.

Bjørn Lomborg. *The Skeptical Environmentalist: Measuring the Real State of the World*. Cambridge University Press, 2001.

Roland Longchamp. *Commande Numérique de Systèmes Dynamiques*. PPUR, 2006.

Matthew Luckiesh and Frank K. Moss. *The Science of Seeing*. D. Van Nostrand Company, Inc., 1937.

Ardeshir Mahdavi. Simulation-based control of building systems operation. *Buildings and Environment*, 36:789–796, 2001.

John Mardaljevic. *Daylight simulation: validation, sky models and daylight coefficients*. PhD thesis, De Montfort University, Leicester, 1999.

Christophe Marty and Marc Fontoynont. Design tool: Algorithms for simultaneous energy and comfort optimisation baed on typical weather data, to be used in building planning. Technical report, Ingélux, Lyon, 2006.

Christophe Marty, Marc Fontoynont, Jens Christoffersen, Marie-Claude Dubois, Jan Wienold, and Werner Osterhaus. User assessement of visual comfort: review of existing methods. Technical report, Ingelux, Lyon, 2003.

METEONORM-2006. Meteonorm 5.x, 2006. URL `http://www.meteotest.ch/en/mn_home`. Last checked 20 October 2006.

Bertrand Meyer. *Object-Oriented Software Construction*. Prentice Hall PTR, second edition edition, 1997.

Joseph J. Michalsky. The *astronomical almanac*'s algorithm for approximate solar position (1950–2050). *Solar Energy*, 40(3):227–235, 1988.

Laurent Michel. *Méthode experimentale d'évalutation des performances lumineuses de bâtiments*. PhD thesis, EPFL, 1999.

P. Moon and D. E. Spencer. Illumination from a nonuniform sky. *Illum. Eng.*, 37(12):707–726, 1942.

Stephan Morgenthaler. *Introduction à la Statistique*. Presses Polytechniques et Universitaires Romandes, 2001.

E. Muck and H. W. Bodmann. Die Bedeutung des Beleuchtungsniveaus bei praktischer Sehtätigkeit. *Lichtechnik*, 13(10):502–507, 1961.

MySQL Development Team. *MySQL Reference Manual*, 2004.

A. Nabil and J. Mardaljevic. Useful daylight illuminance: a new paradigm for assessing daylight in buildings. *Lighting Research and Technology*, 37(1):41–59, 2005.

NAS-2001. Climate change science: an analysis of some key questions. Technical report, National Academy Press, Washington D.C., 2001.

J. Nemecek and E. Grandjean. Results of an ergonomic investigation of large-space offices. *Human Factors*, 15(2):111–124, 1973.

Naomi Oreskes. The scientific consensus on climate change. *Science*, 306:1686, 2004.

Bernard Paule. Private communication, 2006.

Richard Perez et al. All-weather model for sky luminance distribution—preliminary configuration and validation. *Solar Energy*, 50(3):235–245, 1993.

J. R. Petit et al. Climate and atmospheric history of the past 420 000 years from the Vostok ice core, Antarctica. *Nature*, 399:429–436, 1999.

William H. Press et al. *Numerical Recipes in C++*. Cambridge University Press, 2002.

Mark S. Rea, editor. *The IESNA Lighting Handbook, Ninth Edition*. Illuminating Engineering Society of North America, 2000.

Mark S. Rea, editor. *Lighting control: proceedings of the CEA/DBR symposium, Ottawa*, June 1984a.

Mark S. Rea. Window blind occlusion: a pilot study. *Buildings and Environment*, 19(2): 113–137, 1984b.

Christoph F. Reinhart. DAYSIM, 2006. URL `http://irc.nrc-cnrc.gc.ca/ie/lighting/daylight/daysim_e.html`. Last checked 20 October 2006.

Christoph F. Reinhart. *Daylight availability and manual lighting control in office buildings—simulation studies and analysis of measurements*. PhD thesis, University of Karlsruhe, 2001.

Christoph F. Reinhart and Karsten Voss. Monitoring manual control of electric lighting and blinds. *Lighting Research and Technology*, 35(3):243–260, 2003.

W. Riemenschneider. Beleuchtungsstärken für Arbeiträume. *Bull. SEV*, 58(1):19–25, 1967.

A. Robertson et al. Hypothesized climate forcing time series for the last 500 years. *Journal of Geophysical Research*, 106(D14):14 783–14 803, 2001. Data available from `http://www.ncdc.noaa.gov/paleo/pubs/robertson2001/robertson2001.html`. Last checked 15 March 2007.

Darren Robinson and Andrew Stone. Solar radiation modeling in the urban context. *Solar Energy*, 11:295–309, 2004.

Darren Robinson and Andrew Stone. Internal illumination prediction based on a simplified radiosity algorithm. *Solar Energy*, 80:260–267, 2006.

Gary Robinson. A statistical approach to the spam problem. *Linux Journal*, 107, March 2003.

F. Rubinstein, M. Karayel, and R. Verderber. Field study on occupancy scheduling as a lighting management strategy. *Light. Des. Appl.*, 14(5):34–38, 40–45, 1984.

George Sakkis. Learning how to tell ham from spam. *Crossroads, the ACM student magazine*, 11(2), 2004.

Sylvain Sardy. Private communication, 2005.

J. E. Saunders. The role of the level and diversity of horizontal illumination in an appraisal of a simple office task. *Lighting Research and Technology*, 1(1), 1969.

L. J. Schipper, R. Haas, and C. Sheinbaum. Recent trends in residential energy use in OECD countries and their impact on carbon dioxide emissions: a comparative analysis of the period 1973–1992. *Mitigation and Adaptation Strategies for Global Change*, 1:167–196, 1996.

P. Kenneth Seidelmann, editor. *Explanatory Supplement to the Astronomical Almanac*. University Science Books, Mill Valley, California, 1992.

SIMBAD-2006. Simbad, 2006. URL `http://kheops.champs.cstb.fr/Simbadhvac/`. Last checked 20 October 2006.

Bojana Spasojević and Ardeshir Mahdavi. Sky luminance mapping for computational daylight modeling. In *Proceedings of the Ninth International IBPSA Conference, Montréal, Canada*, pages 1163–1169, 2005.

Yannick Sutter, Dominique Dumortier, and Marc Fontoynont. The use of shading systems in VDT task offices: a pilot study. *Energy & Buildings*, 38(7):780–789, 2006.

G. Söllner. Über angenehme Leuchtdichteverhältnisse in der Innenraumbeleuchtung. Laborbericht 1.14/22, 1966.

Teolan Tomson and Gunnar Tamm. Short-term variability of solar radiation. *Solar Energy*, 2006.

P.R. Tregenza. Subdivision of the sky hemisphere for luminance measurements. *Lighting Research and Technology*, 19:13–14, 1987.

P.R. Tregenza and I.M. Waters. Daylight coefficients. *Lighting Research and Technology*, 15: 65–71, 1983.

N. Tuaycharoen and P.R. Tregenza. Discomfort glare from interesting images. *Lighting Research and Technology*, 37(4):329–341, 2005.

William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S*. Springer Verlag, 2002.

E. Vine, E. Lee, R. Clear, D. DiBartolomeo, and S. Selkowitz. Office worker response to an automated venetian blind and electric lighting system: a pilot study. *Energy and Buildings*, 28(2):205–218, 1998.

Ann R. Webb. Considerations for lighting in the built environment: non-visual effects of light. *Energy and Buildings*, 38(7):721–727, 2006.

J. M. Westhoff and H. W. Horeman. Kriterien für optimale Beleuchtungsniveaus. *Lichttechnik*, 15(1):24–26, 1963.

H. C. Weston. *The relation between illumination and visual efficiency: the effect of size of work*. H. M. Stationery Office, 1935.

H. C. Weston. *The relation between illumination and visual efficiency: the effect of brightness contrast*. H. M. Stationery Office, 1945.

Michael Wetter and Jonathan Wright. A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. *Building and Environment*, 39:989–999, 2004.

Jan Wienold and Jens Christoffersen. Towards a new daylight glare rating. In *Lux Europa 10th European Lighting Conference*, 2005.

Jan Wienold and Jens Christoffersen. Evaluation methods and development of a new glare prediction model for daylight environments with the use of ccd cameras. *Energy and Building*, 38(7):743–757, 2006.

WIKI-GCD. Great-circle distance. URL `http://en.wikipedia.org/wiki/Great-circle_distance`. Last checked 7 February 2007.

WIKI-PRT. Pearl river tower. URL `http://en.wikipedia.org/wiki/Pearl_River_Tower`. Last checked 11 May 2007.

L. T. Wong, K. W. Mui, N. K. Fong, and P. S. Hui. Bayesian adaptive comfort temperature (BACT) of air-conditioning system in subtropical climate. *Building and Environment*, 42: 1983–1988, 2007.

Jonathan A. Wright, Heather A. Loosemore, and Raziyeh Farmani. Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and Buildings*, 34: 959–972, 2002.

David Lindelöf

Chemin de l'Ancienne-Ecole 37, 1288 Aire-la-Ville
30, married
+41 (0)79/415.66.41
lindelof@ieee.org

Education

Swiss Federal Institute of Technology in Lausanne (EPFL) — Lausanne, CH
Doctorate: 2003–present, expected September 2007

- Doctoral courses taken: Probabilities, Financial Statistics, Mathematica for Materials Engineers, Solar Radiation and Daylight Modeling, Scientific Communication.

Swiss Federal Institute of Technology in Lausanne (EPFL) — Lausanne, CH
Master in Physics: 1993–2000

Professional Experience

Adhoco AG — Winterthur, CH
Product developer: July 2004–present

- Built the OSGi-based software for an intelligent embedded home automation controller.
- Designed a web-based home automation solution through the corporate website.
- Managed a geographically distributed development effort.

Visnet — Geneva, CH
Co-founder: October 2003–June 2004

- Designed and installed a KNX home automation solution for a house owner.
- Simulated financial investment strategies with quantum random numbers.
- Developed a website for a real-estate agent.
- Did IT support for a financial firm.

Solar Energy and Building Physics Laboratory — Lausanne, CH
PhD candidate: February 2003–present

- Doctoral work: developed a daylight modeling algorithm for embedded controllers and an intelligent daylight control algorithm.
- Did several statistical studies on the behaviour of an administrative's building's occupants.

CERN, University of Zürich — Geneva, CH
Research assistant: April 2000–January 2003

- Assembled and calibrated elements of the antimatter detector of the Athena experiment.
- Programmed several C++ routines of the event reconstruction module.

ABB Corporate Research — Baden, CH
Trainee: summer 1998

- Measured and modeled the aging of polymeric film capacitors.
- Simulated lifetimes of capacitors under realistic conditions.

Skills

Technical project management:
- Requirements gathering, Cost and effort estimation methods, Coaching, Software Design and Architecture, Development Tools and Infrastructure.

Mathematics:
- Probabilities, Statistics, Financial Models, Stochastic Models, Bayesian Networks, Monte-Carlo, Game Theory.

Physics:
- Building Physics, Daylight Modeling, Thermodynamics, Classic and Quantum Electrodynamics.

Operating Systems:
- Linux (Mandrake, Slackware, SUSE, Gentoo, Debian, Knoppix, Ubuntu, µClinux), Windows 98/2000/XP/Vista.

Programming Languages:
- Proficient in C/C++, Java, Python, Perl, R/S, LaTeX, UNIX Shell.
  Familiar with Lisp, SQL, PHP.

Scientific Software:
- Proficient in R, Mathematica, MATLAB, Simulink, Simbad, Igor Pro.
  Familiar with LabVIEW, Radiance.

Tools and Systems:
- Proficient in OSGi, CVS/RCS, Subversion, MySQL, Ant, Emacs, Qmail.
  Familiar with NFS, Apache, Make, Samba.

Others:
- Control Theory, System Identification, Decision Theory, Machine Learning, Bayesian Artificial Intelligence.

Languages

French: mother tongue (C2).

Swedish: mother tongue (C2).

English: fluent (C2).

German: conversant (B2).

Japanese: notions (A2).

Achievements and Activities

- Member of the IEEE Computer Society.
- Member of the Association for Computing Machinery.
- Member of the International Building Performance Simulation Association, Swiss Affiliate (IBPSA-CH).
- Administrator of the open-source Java GNU Scientific Library project (`http://sf.net/projects/jgsl/`).
- Author of the Computing and Smart Buildings weblog (`http://www.visnet.ch/smartbuildings/`).
- Vice-president (public relations) of the EPFL-UNIL Toastmasters Club.
- Former president of the CERN Chess Club.
- Hobbies: Conflict Simulations, Scuba diving, Travel, Photography.

Refereed journal papers, conference proceedings and seminars

As main author

- D. Lindelöf: "A simplified daylight model suitable for daylight controllers", *Solar Energy* (submitted)
- D. Lindelöf and N. Morel: "Bayesian estimation of visual discomfort", *Building Research and Information* (accepted for publication)
- D. Lindelöf: "Integrating R in an advanced building control system", *useR! Conference, Vienna*, 2006
- D. Lindelöf and N. Morel: "A field investigation of the intermediate light switching by users", *Energy and Buildings* 38 nr. 7, 2006
- D. Lindelöf, A. Guillemin, L. Wilhelm et R. Altherr: "aHeart: a hardware-independent building control API", *Proceedings of the CISBAT Conference, Lausanne*, 2005
- D. Lindelöf and N. Morel: "Bayesian optimization of user visual comfort", *Proceedings of the Lux Europa 2005 Conference, Berlin*, 2005
- "Contrôle bio-mimétique et adaptatif d'un store vénitien (projet Ecco-Build)", seminar given at the Solar Energy and Building Physics Laboratory on 13th February 2003
- "Experimental searches for CPT violation", seminar given at the University of Zürich on 5th December 2001 and at CERN on 7th February 2002

As coauthor

- M. Amoretti et al.: "High rate production of antihydrogen", *Phys. Lett.* B 578, 2004
- M. Amoretti et al.: "The ATHENA antihydrogen apparatus", *Nucl. Inst.Meth. Phys. Res.* A 518, 2004
- M. Amoretti et al.: "Production and detection of cold antihydrogen atoms", *Nature* 419, pp. 456, 2002
- C. Amsler et al.: "Temperature Dependence of Pure CsI: Scintillation Light Yield and Decay Time", *Nucl. Inst. Meth. Phys. Res.* A 480, pp. 494, 2002